A TRIDENT SCHOLAR PROJECT REPORT

NO. 223

"Computational Solutions to the Protein Folding Problem"

UNITED STATES NAVAL ACADEMY
ANNAPOLIS, MARYLAND

U.S.N.A. - Trident Scholar project report; no. 223 (1994)

"Computational Solutions to the Protein Folding Problem"

by
Midshipman Vann H. Walke, Class of 1994
U.S. Naval Academy
Annapolis, Maryland

Advisor: Associate Professor Andrew T. Phillips
Computer Science Department

Accepted for Trident Scholar Committee

Chair

19 May 1994

Date

REPORT DOCUMENTATION PAGE

Form Approved
OMB no. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	19 May 1994	

4. TITLE AND SUBTITLE
Computational solutions to the protein folding problem

5. FUNDING NUMBERS

6. AUTHOR(S)
Vann H. Walke

7. PERFORMING ORGANIZATIONS NAME(S) AND ADDRESS(ES)
U.S. Naval Academy, Annapolis, MD

8. PERFORMING ORGANIZATION REPORT NUMBER
USNA
Trident Scholar project report; no. 223 (1994)

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Accepted by the U.S. Trident Scholar Committee

12a. DISTRIBUTION/AVAILABILITY STATEMENT
This document has been approved for public release; its distribution is UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
The protein folding problem attempts to predict the native, or folded, state of a protein in three-dimensional space, given its primary sequence of amino acids. One common approach for a solution is to treat each complex amino acid as a single sphere, or "united atom," and to model each peptide linkage between residues by a virtual bond between spheres. Computational efforts being examined rely on two major assumptions: 1) for any specific molecular conformation, a corresponding potential energy function can be computed, and 2) the three-dimensional, folded state corresponds to the global minimum of this energy function. The optimization method being used to minimize the potential energy involves collecting a large number of conformers, each attained by finding a local minimum of the potential energy function from a random starting point. The information from these conformers is then used to form a convex quadratic global underestimating function for the potential energy of the known conformers. The minimum of this underestimator is used to predict the global minimum for the function, allowing a localized conformer search to be performed based on the predicted minimum. The new set of conformers generated by the localized search can serve as the basis for another quadratic underestimation. After several repetitions, the global minimum can be found with reasonable assurance. The conformer which lies at the global minimum represents the three-dimensional folded state of the molecule.

14. SUBJECT TERMS	15. NUMBER OF PAGES
protein folding problem, global underestimator, conformer, mathematical models	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED

Computational Solutions to the Protein Folding Problem

V. H. Walke

Abstract

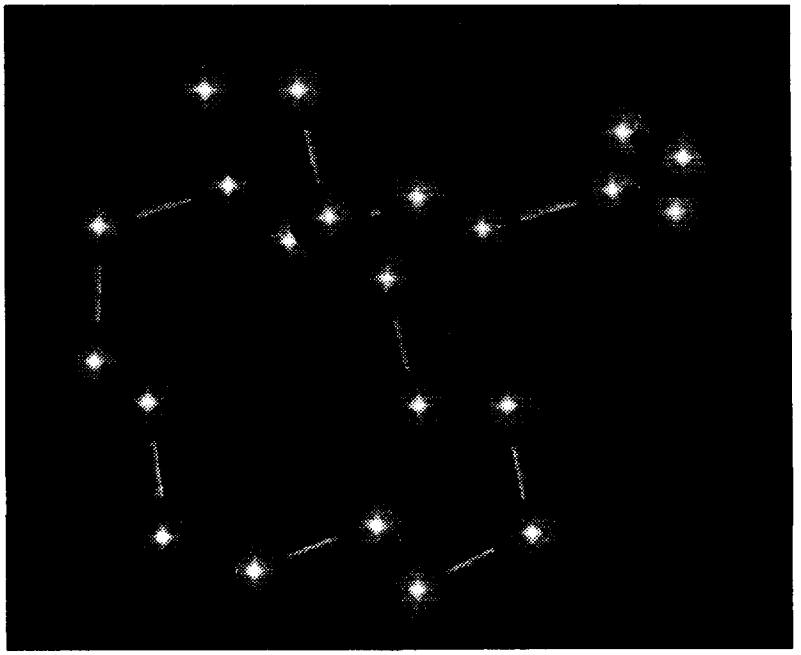
The protein folding problem attempts to predict the native, or folded, state of a protein in three-dimensional space, given its primary sequence of amino acids. One common approach for a solution is to treat each complex amino acid as a single sphere, or "united atom," and to model each peptide linkage between residues by a virtual bond between spheres. Computational efforts being examined rely on two major assumptions: 1) for any specific molecular conformation, a corresponding potential energy function can be computed, and 2) the three-dimensional, folded state corresponds to the global minimum of this energy function. The optimization method being used to minimize the potential energy involves collecting a large number of conformers, each attained by finding a local minimum of the potential energy function from a random starting point. The information from these conformers is then used to form a convex quadratic global underestimating function for the potential energy of the known conformers. The minimum of this underestimator is used to predict the global minimum for the function, allowing a localized conformer search to be performed based on the predicted minimum. The new set of conformers generated by the localized search can serve as the basis for another quadratic underestimation. After several repetitions, the global minimum can be found with reasonable assurance. The conformer which lies at the global minimum represents the three-dimensional folded state of the molecule.

Preface

I offer thanks to all the people who helped and supported me throughout this project. In particular I would like to express my deepest thanks to the Computer Science Department of the United States Naval Academy for their constant support, to Dr. Robert Artigiani, who helped expand my educational horizons and was always there to provide encouragement, and most significantly to Dr. Andrew T. Phillips. His enthusiasm, drive, and ideas kept me going and most importantly going in the right direction. Without their help this report would not have been possible.

Contents

List of Tables and Figures



I. Introduction

A. The Protein Folding Problem

One of the most important and most difficult problems in biophysics and biochemistry is the protein folding problem. In the late 1950's Christian B. Anfinsen and his colleagues at the National Institute of Health made the remarkable discovery that the folded state of a protein was completely dependent on the one-dimensional linear sequence (i. e. "primary" sequence) of amino acids from which the protein is constructed. External factors, such as enzymes present at the time of folding, have no effect on the final or native state of the protein. This led to the formulation of the protein folding problem: Given a known primary sequence of amino acids, what would be its native, or folded, state in 3-dimensional space? Despite over thirty years of research, a solution to the protein folding problem remains elusive (Richards 1991).

A solution to the protein folding problem would provide immeasurable benefits to the biotechnology field. Many new products in the biotechnology industry are novel proteins. It is already possible to design "custom" amino acid sequences quickly with DNA sequence analysis; however, the inability to predict the folded state of the protein, which determines the protein's functionality, forces biotechnology industries to rely on long, expensive methods of development. When Eli Lilly made his first attempts at producing insulin for diabetes, a biologically inactive material resulted, which resembled scrambled eggs. While he was able to splice the human gene for insulin into DNA molecules of bacteria and get them to synthesize the needed protein chain, incorrect folding caused the process to fail. Only after considerable expenses in time and money was the correct folded sequence found. Currently, the three-dimensional folded state of a protein can only be

found using X-ray crystallography, an enormously expensive and time consuming process which can take as much as several months or even years.

While methods have been developed which attempt to predict a folded protein's secondary structure (distinct sections having characteristic shapes including helices, beta strands, and turns), these methods rarely yield information which provides a description of the tertiary structure (the folded state). As a result, the protein folding problem has generally been thought to be, if not completely insoluble, not likely to be solved for a very long time.

B. Possible Solutions to the Protein Folding Problem

Recent events have renewed hope for a solution to the problem. Several successful predictions have been made and announced before the experimental structures were known. While most of these predictions have been made with a blend of a human expert's abilities and computer assistance, fully automated methods have shown promise for producing previously unattainable accuracy. Advances in theory, experiment, and computing power, coupled with growing interest from industry, have kindled new optimism for a solution to the protein folding problem (Richards 1991, Benner and Gerloff 1993).

Computational attempts to solve the folding problem can be divided into two basic strategies: human based and machine based predictions. The first relies on a close interaction between a human expert and a computer model which simulates the behavior of the protein. One such system was developed by Mark C. Surles at the University of North Carolina at Chapel Hill. The graphical modeling system he developed, "Sculpt", provides an environment in which an expert can graphically manipulate the protein into a folded state. Through the use of the Sculpt program, the expert can insure that the protein

structure complies with established biochemical rules (Surles 1992). While this approach is a huge improvement over a purely human based analysis of protein structure, it still remains dependent on possibly enormous amounts of time spent by an expert.

Machine based prediction strategies attempt to lessen the reliance on experts by developing a completely computational method. Such approaches are generally based on two assumptions. First, there exists a potential energy function for the protein; and second, the folded state corresponds to the structure with the lowest potential energy (minimum of the potential energy function) and is thus in a state of thermodynamic equilibrium. This view is supported by in-vitro observations that proteins can successfully refold from a variety of denatured states. The alternate assumption is that the native state of the protein may be at a local energy minimum which can be separated from the true global energy minimum by a large kinetic barrier. Recent studies of the α–Lytic protease revealed two conformational forms, active and inactive, separated by just such a kinetic barrier (Baker et al. 1992). While the active, or stable, form of the molecule occurred at the global minimum, a catalyst is required to change the "molten globule" found at the inactive state into the final state for the protein. This evidence seems to support the view that the folded state may not lie at the global minimum in every case. Ruben A. Abagyan raises two questions related to this phenomenon (Abagyan 1993):

(1) Could a metastable state have all the features of the normal protein?

(2) If it could, is this behavior typical for all proteins?

Currently there is no clear evidence for a positive answer to the first question. The α–Lytic experiments, while raising important questions, do not provide evidence concerning the properties of the normal protein (the final state for α-Lytic occurred at the global mini-

mum). The second question can be answered definitively in the negative. Small single-domain proteins do fall into their lowest energy state. Evolutionary theory also supports a folded state at a global energy minimum. Protein sequences have evolved under pressure to perform certain functions, which for most known occurrences require stable, unique and compact structures. Unless specifically required for a certain function, there was no biochemical need for proteins to hide their global minimum behind a large kinetic energy barrier. While kinetic blocks may occur, they should be limited to special proteins developed for certain functions, or occur randomly with the probability of kinetic blocks growing with problem size (Abagyan 1993).

C. Models for Computational Methods

Unfortunately, finding the "true" energy function of a protein, if one even exists, is so unimaginably complicated that there are no means nor hope of determining it exactly. With proteins ranging in size up to 1,053 amino acids (a collagen found in tendons), and covering an almost infinite number of possible conformations, exhaustive conformational searches will never be possible.

One possible way of finding the global energy minimum is to use a simplified model of the protein system. By using a simplified model, the complexity of the problem formulation could be reduced to an acceptable level for optimization techniques. Care must be taken, however, to insure that the error included in such an approximation does not drive the computational solution too far away from the true native state. Simplified computational protein models can be currently divided into two major classes: lattice restricted models and models in continuous space.

The first of these methods, lattice models, relies on constraining the position of the

amino acids to the vertices of a three-dimensional lattice. By limiting the possible positions for amino acid placement, lattice methods have the potential for large increases in algorithm speed. This increase, however, is reliant on the quality of the grid developed. A lattice with a very large number of vertices could provide an accurate model of the protein. Unfortunately, this increase in accuracy carries a corresponding decrease in speed. Conversely, a very coarse lattice with relatively few vertices enables fast execution time, but may not provide an accurate solution.

The second class of techniques attempts to avoid the reliance on lattice restrictions by performing the optimization in continuous space. These methods represent a protein as a string of beads (amino acids) whose position is defined by their location relative to the previous three beads. One significant advantage of this formulation of the folding problem is that it allows the model to take advantage of known scientific knowledge about the chemical construction of proteins. The use of knowledge such as the Ramachandran plot (Lehninger 1970) which specifies the possible values for certain angles in a protein, further simplifies the problem. This class of problems is also appealing since the problems may be subjected to a variety of global optimization methods including minimization from random starting conformations, molecular dynamics, Monte Carlo techniques, and simulated annealing (Abagyan 1993).

This paper will examine several possibilities for the global optimization of a protein's potential energy function. The approaches rely on two major assumptions:

1. For any specific molecular conformation, a corresponding potential energy can be computed from a potential energy function model of the protein.
2. The native state corresponds to the global (or near global) minimum of this

energy function.

Each amino acid is represented as a single point whose position is determined by a bond length, bond angle, and torsion angle triple. Several different methods for this optimization, including both continuous and lattice based approaches, will be examined.

II. The Potential Energy Function

A. Molecular Representations

Molecular structure information is generally given in terms of bond lengths, bond angles, and torsion angles. A protein consists of n amino acids in the "primary" sequence, $a_1,a_2,...a_n$, where a_i represents the i^{th} amino acid in the sequence. For every consecutive pair of amino acids, a_{i-1} and a_i, l_i represents the distance by which they are separated. For every three consecutive amino acids, a_{i-2}, a_{i-1}, a_i, the bond angle, θ_i, represents the position of the third amino acid with respect to the line containing the previous two amino acids. Similarly, for every four consecutive amino acids, a_{i-3}, a_{i-2}, a_{i-1}, and a_i, the torsion angle, φ_i, represents the relative position of the fourth amino acid, a_i, with respect to the plane containing the previous three amino acids (See Figure 2.1).

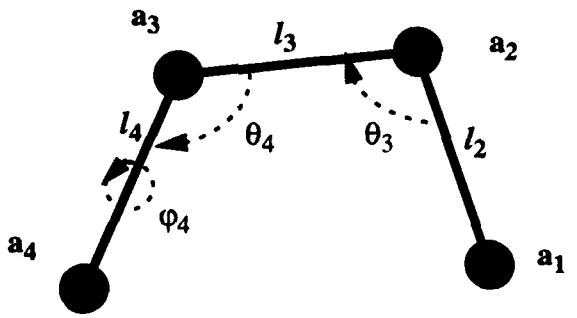


Figure 2.1 Conformation for a Four Amino Acid Sequence

B. Potential Energy Function

Empirical representations of the protein's potential energy include energy terms to represent chemical bonds, angles, and torsions, as well as non-bonded interactions between amino acids farther apart in the chemical structure. This simplified definition resembles a physical model in which beads (amino acids) are connected by springs (first

term of Equation 2.1) at a distance of l_i. Bond angles, θ_i, which are determined by a sequence of three amino acids (a_{i-2},a_{i-1},a_i) are maintained by similar "springs" (the second term). Torsion angles, φ_i, are modeled by a trigonometric based penalty (third term of Equation 2.1). Such a formulation is often called a molecular mechanics potential. Therefore, one formulation frequently shown to express the overall potential is shown in Equation 2.1, where k_l and k_θ are the bond stretching and angle bending force constants respectively. In the third term of the equation, $V_{\bar{n}}$ is the $\bar{n}$-fold torsional constant with a phase shift of δ. This term provides for preferred torsion angles for the protein. For example, if $n = 3$ and the phase shift, δ, is zero, preferred torsion angles (configurations resulting in a small penalty) can be found at 60°, 180°, and 300°. (Since the cosine of 180° or any odd multiple of 180° is -1, the penalty incurred by the preferred torsions will be zero.) The constants, l_0 and θ_0 represent the preferred bond length and bond angle for each set of amino acids.

$$U = \sum_i \frac{1}{2} k_l (l_i - l_0)^2 + \sum_i \frac{1}{2} k_\theta (\theta_i - \theta_0)^2 + \sum_i \frac{V_{\bar{n}}}{2} [1 + \cos(\bar{n}\varphi_i + \delta)] \qquad \text{Equation 2.1}$$
$$+ \sum_{i,l} \varepsilon_{i,l} \left(\left(\frac{\sigma_{i,l}}{r_{i,l}} \right)^{12} - 2 \left(\frac{\sigma_{i,l}}{r_{i,l}} \right)^{6} \right).$$

The final portion of the equation, $\sum_{i,l} \varepsilon_{i,l} \left(\left(\frac{\sigma_{i,l}}{r_{i,l}} \right)^{12} - 2 \left(\frac{\sigma_{i,l}}{r_{i,l}} \right)^{6} \right)$, is known as the Lennard-Jones pairwise potential. This term defines the potential energy contributions of amino acids separated by more than two residues along the primary chain. Graphically, the Lennard-Jones term can be represented as a plot (Figure 2.2) of the potential energy of the protein (in kcal) versus the distance (in Å) separating the amino acid residues in the pro-

tein. In Equation 2.1, $\varepsilon_{i,l}$ and $\sigma_{i,l}$ are the Lennard-Jones coefficients, which are constants

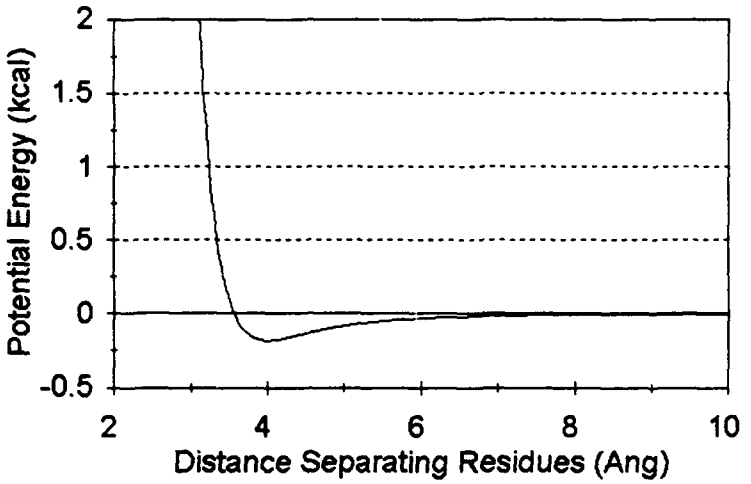


Figure 2.2 Lennard-Jones Pairwise Potential (ε=0.181, σ=4.0)

defined by the relationships between the two specific beads (amino acids) involved. The $r_{i,l}$ terms in the Lennard-Jones expression represent the Euclidean distance between the amino acids a_i and a_l (Ferguson et al. 1994) (Case 1993).

III. Conversions

A. Coordinate Representations

Potential energy functions for molecular conformations usually involve computing the distance between molecules. For example, the potential function discussed in Chapter II (page 12) contains the Lennard-Jones pairwise potential, $\sum_{i,l}\varepsilon_{i,l}\left(\left(\frac{\sigma_{i,l}}{r_{i,l}}\right)^{12}-2\left(\frac{\sigma_{i,l}}{r_{i,l}}\right)^{6}\right)$, which uses the distance between all amino acids separated by more than two residues along the primary chain (such as amino acids a_1 and a_4). Unfortunately, computing distances using bond lengths, bond angles, and torsion angles is extremely difficult and very time consuming. Since optimization ***requires*** this computation to be executed often, it is much more profitable to convert the problem into cartesian coordinates before performing the optimization. Once this conversion has been completed, the length between the residues can be found with a minimal amount of computation time. However, the conversion itself becomes very complex since every set of three residues creates *its own reference frame* for bond lengths, bond angles, and torsion angles.

B. The Conversion Geometry

To perform the conversion from a chemical representation (bond lengths, bond angles, and torsion angles) to a cartesian representation, the first three points in the sequence can be fixed, without loss of generality, as depicted in Figure 2.1. The first amino acid, a_1, is fixed at the origin, (0,0,0). The second amino acid, a_2, is positioned at (l_2,0,0), which is a distance from a_1 equal to the bond length (l_2) along the x axis. The location of the third amino acid, a_3, is fixed at ($l_2+l_3\cos(\theta_3)$,$l_3\sin(\theta_3)$,0). The position of the fourth and any other amino acids constituting the primary sequence are found using the cartesian

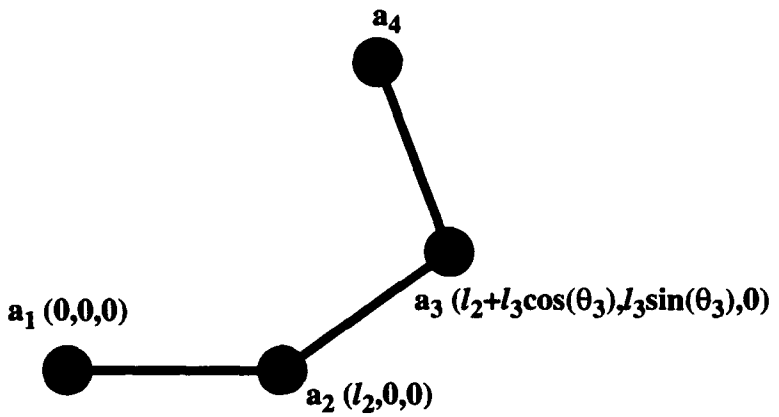


Figure 3.1 Four Amino Acid Sequence (Cartesian Coordinates)

data associated with the previous three points, and the bond length, bond angle, and torsion angle associated with the amino acid under consideration.

The position of a fourth point, a_l, in a four amino acid sequence, a_i, a_j, a_k, a_l, is completely determined by its chemical representation: bond length (l_l), bond angle (θ_l), and torsion angle (φ_l). To convert this information into x, y, and z coordinates, the following quantities must be computed: (1) a plane based on the bond angle and bond length, (2) a plane based on the torsion angle and bond length, and (3) a sphere with radius equal to the bond length. The intersection of the two planes yields a line, which can then be intersected with the sphere to provide two possible positions for the bead a_l. The ambiguity between these two points can be resolved by closer inspection of the definition of the torsion angle. The remaining sections of this chapter serve to illustrate this technique.

C. Plane based on Bond Angle

The bond angle, θ_l, and bond length, l_l, define a plane in which the fourth amino acid, a_l, must lie (See Figure 3.2). The normal vector to this plane is defined as the vector between the two previous amino acids (in Figure 3.2, a_j and a_k). If $\mathbf{r}_k$ represents the position of atom k, and $\mathbf{r}_j$ represents the position of atom j, then:

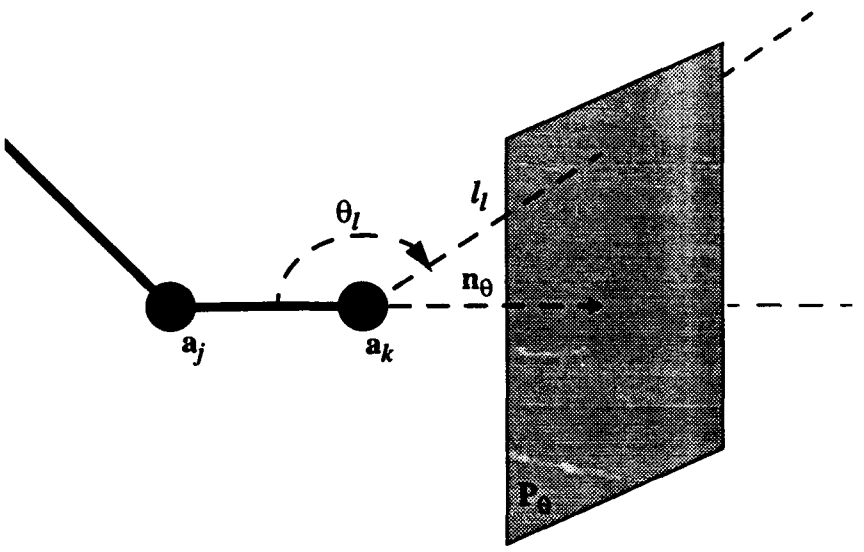


Figure 3.2 Plane Found From Bond Angle and Bond Length

$$\mathbf{r}_{kj} = \mathbf{r}_k - \mathbf{r}_j \qquad \text{Equation 3.1}$$

and this vector, in fact, is the normal vector, $\mathbf{n}_\theta$, to the plane, $\mathbf{P}_\theta$. In addition to a normal vector, a point on the plane must be specified to complete the construction of the plane. The bond angle, when combined with the bond length, provides enough information to compute this point. That is, the distance from the previous point, a_k, to the plane is given by:

$$d = -l_l \cos\theta_l . \qquad \text{Equation 3.2}$$

Since this distance is taken in the direction of the normal vector, a point on the plane is given by:

$$\mathbf{r}_\theta = d\frac{\mathbf{n}_\theta}{\|\mathbf{n}_\theta\|} + \mathbf{r}_k . \qquad \text{Equation 3.3}$$

Therefore, the equation of the plane is:

$$A_\theta x + B_\theta y + C_\theta z - D_\theta = 0 \qquad \text{Equation 3.4}$$

where A_θ, B_θ, and C_θ are the i, j, and k components of the normal vector, $\mathbf{n}_\theta$. The constant term, D_θ, is given by:

$$D_\theta = A_\theta x_\theta + B_\theta y_\theta + C_\theta z_\theta \qquad \text{Equation 3.5}$$

where $(x_\theta, y_\theta, z_\theta)$ is the point $\mathbf{r}_\theta$.

D. Plane based on Torsion Angle

The position of the amino acid, a_l, can be further restricted by a plane, $\mathbf{P}_\varphi$, constructed from the torsion angle and bond length (see Figure 3.4). The normal vector to the plane, $\mathbf{P}_\varphi$, is found by taking the cross product of $\mathbf{r}_{kj}$ and $\mathbf{r}_{ij}$ (for $\mathbf{r}_{kj}$ see Equation 3.1) so

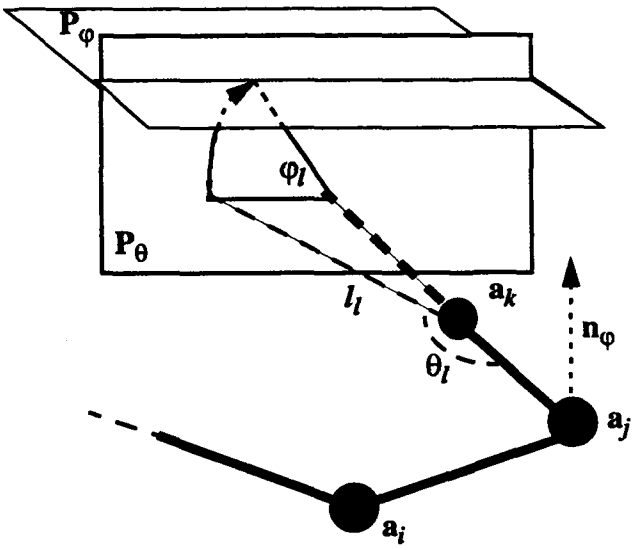


Figure 3.3 Plane Derived From Torsion Angle and Bond Length

that:

$$\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j \qquad \text{Equation 3.6}$$

$$\mathbf{n}_\varphi = \mathbf{r}_{kj} \times \mathbf{r}_{ij}. \qquad \text{Equation 3.7}$$

This plane is will be orthogonal to the first plane, since its normal vector implicitly involves the cross product of the bond angle plane's normal vector ($\mathbf{n}_\theta = \mathbf{r}_{kj}$). A point on the plane, $\mathbf{P}_\varphi$, can be found using the distance from a_k to the plane, $\mathbf{P}_\varphi$, and the known normal vector, $\mathbf{n}_\varphi$. The distance to the plane is obtained by:

$$d = l_l \sin(180° - \theta_l) \sin(\varphi_l) . \qquad \text{Equation 3.8}$$

The point, $\mathbf{r}_\varphi$, on this plane is then:

$$\mathbf{r}_\varphi = d\frac{\mathbf{n}_\varphi}{\|\mathbf{n}_\varphi\|} + \mathbf{r}_k . \qquad \text{Equation 3.9}$$

As in section C, the equation for the plane is given by:

$$A_\varphi x + B_\varphi y + C_\varphi z - D_\varphi = 0 \qquad \text{Equation 3.10}$$

where A_φ, B_φ, and C_φ are the i, j, and k components of the normal vector, $\mathbf{n}_\varphi$. The constant term, D_φ, is given by:

$$D_\varphi = A_\varphi x_\varphi + B_\varphi y_\varphi + C_\varphi z_\varphi \qquad \text{Equation 3.11}$$

where $(x_\varphi, y_\varphi, z_\varphi)$ is the point $\mathbf{r}_\varphi$.

E. Bond Length Sphere

Finally, a sphere of radius equal to the bond length about the last point, a_k, is described. The equation is simply:

$$(x - x_k)^2 + (y - y_k)^2 + (z - z_k)^2 = l_l^2 \qquad \text{Equation 3.12}$$

where (x_k, y_k, z_k) is the cartesian position of the point a_k.

F. Line at Intersection of Bond Angle and Torsion Angle Planes

Since the bond angle plane and torsion angle plane are orthogonal and therefore guaranteed to intersect, the line formed by this intersection can be found. The direction of this line, its line vector, is:

$$l = \mathbf{n}_\theta \times \mathbf{n}_\varphi . \qquad \text{Equation 3.13}$$

The cartesian components of this line vector are denoted by, (A_l, B_l, C_l). To complete the line definition, a point on the line must be known. This can be easily accomplished by finding any solution to the system of two equations defining the two planes, $\mathbf{P}_\theta$ and $\mathbf{P}_\varphi$:

$$\begin{bmatrix} A_\theta & B_\theta & C_\theta \\ A_\varphi & B_\varphi & C_\varphi \end{bmatrix} \begin{bmatrix} x_{pt} \\ y_{pt} \\ z_{pt} \end{bmatrix} = \begin{bmatrix} D_\theta \\ D_\varphi \end{bmatrix} . \qquad \text{Equation 3.14}$$

By setting any single variable to zero (since the system is underdetermined) a point, $\mathbf{r}_{pt} = (x_{pt}, y_{pt}, z_{pt})$, which lies on the line can be obtained. Therefore, using a parametric formulation (with t representing the parameter), the line is:

$$x = x_{pt} + A_l t \qquad \text{Equation 3.15}$$

$$y = y_{pt} + B_l t \qquad \text{Equation 3.16}$$

$$z = z_{pt} + C_l t . \qquad \text{Equation 3.17}$$

G. Intersection of Line and Sphere

By combining the information from the line found in the previous section and the sphere based on bond length, the position of the new residue can be limited to two possible points (Figure 3.4). Substituting the line equations into the sphere equation developed

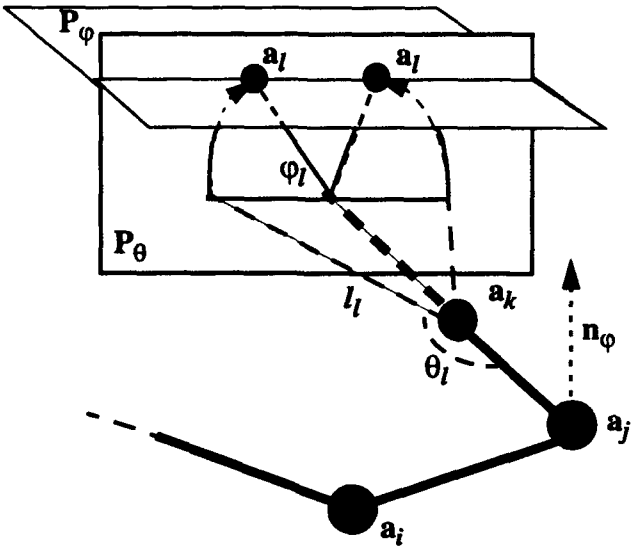


Figure 3.4 Possible Locations for a_l

in Equation 3.12, one can derive the following relationships:

$$(x_{pt} + A_l t - x_k)^2 + (y_{pt} + B_l t - y_k)^2 + (z_{pt} + C_l t - z_k)^2 = l_l^2. \qquad \text{Equation 3.18}$$

Since t is the only unknown, by the quadratic rule,

$$t = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \qquad \text{Equation 3.19}$$

where,

$$a = \left(A_l^2 + B_l^2 + C_l^2\right) \qquad \text{Equation 3.20}$$

$$b = 2\left[A_l(x_{pt} - x_k) + B_l(y_{pt} - y_k) + C_l(z_{pt} - z_k)\right] \qquad \text{Equation 3.21}$$

$$c = (x_{pt} - x_k)^2 + (y_{pt} - y_k)^2 + (z_{pt} - z_k)^2. \qquad \text{Equation 3.22}$$

The two possible points can be easily found by solving Equation 3.15 through Equation 3.17 using the two values of t derived above.

H. Resolving to One Point

The ambiguity for the two points can be resolved by simply testing the resultant torsion angles against the desired torsion angle, φ_l. One of the two points is selected and its torsion angle is computed. If this torsion angle is within a small error of the desired torsion angle, φ_l, the point is accepted. Otherwise, the point is discarded and the other point defined by the intersection is selected. Hence, to complete the computation of the torsion angle three vectors, $\mathbf{r}_{ij}$ (Equation 3.6), $\mathbf{r}_{kj}$ (Equation 3.1), and $\mathbf{r}_{lk}$, are defined:

$$\mathbf{r}_{lk} = \mathbf{r}_l - \mathbf{r}_k. \qquad \text{Equation 3.23}$$

Two new vectors are constructed: $\mathbf{t}$, which is normal to the plane defined by a_i, a_j, and a_k, and $\mathbf{u}$, which is normal to the plane defined by a_j, a_k, a_l:

$$\mathbf{t} = \mathbf{r}_{ij} \times \mathbf{r}_{kj} \qquad \text{Equation 3.24}$$

$$\mathbf{u} = \mathbf{r}_{lk} \times \mathbf{r}_{kj}. \qquad \text{Equation 3.25}$$

The torsion angle, φ, is defined as the angle between the two planes passing through a_i, a_j, and a_k, and a_j, a_k, a_l, respectively. Therefore:

$$\varphi = \operatorname{acos}\left(\frac{\mathbf{t} \cdot \mathbf{u}}{\|\mathbf{t}\| \|\mathbf{u}\|} \right). \qquad \text{Equation 3.26}$$

When $\mathbf{t}$ and $\mathbf{u}$ are parallel, $\varphi = 0$. If $\mathbf{t}$ and $\mathbf{u}$ are antiparallel, then $\varphi = 180^\circ$ (Swope and Ferguson 1992). This value of φ is compared, as described above, to φ_l to eliminate any ambiguity and therefore conclude the computation of the cartesian representation of the point a_l (for another method of computing the conversions, see Maranas and Floudas 1994).

I. Effectiveness of Conversion Method

The conversion method developed executes in $O(n)$ steps, since every amino acid

position must be evaluated. The overhead incurred by such a conversion however, allows the computation of the distances (and the Lennard-Jones term) between all pairs of amino acids in $O(n^2)$ steps, since there are n^2 pairs (length computations are $O(1)$ in cartesian coordinates). In comparison, computing the distance *directly* from the protein's chemical representation would incur an $O(n^3)$ algorithm, since each length computation would incur an $O(n)$ penalty. The additional overhead of maintaining a separate copy of the amino acid in cartesian coordinates is more than compensated for in decreased execution time.

IV. *n*-Chain Hydrocarbons: A Sample Problem

A. *n*-Chain Hydrocarbon Problem

Techniques aimed at solving the protein folding problem can be tested on simplified versions of the problem to provide insight into the method's particular characteristics and potential for a solution. One such simplified version is known as the *n-chain hydrocarbon problem*. This problem attempts to find the 3-dimensional conformation of a chain of n hydrocarbons.

Hydrocarbons can be modeled in a manner similar to proteins: a string of beads (a_i, where $i = 1,...,n$), positioned by their respective bond lengths, l_i, bond angles, θ_i, and torsion angles, φ_i (see Chapter II). Each "residue" in this case, however, can be considered *identical*. Scientific knowledge has determined that all hydrocarbons have "preferred" bond lengths and bond angles. This information can be used in some optimization techniques to reduce the number of problem parameters. Note that similar information has been developed for proteins using the Ramachandran plot (Lehninger 1970).

B. The Conformation of Heptane

Heptane is an n-chain hydrocarbon of length seven ($n = 7$). It provides a simple, tractable problem for which the conformational space can (almost) be exhaustively explored. While this structure reflects great simplifications over the protein folding problem, its complexity must not be underestimated.

Using the potential energy function (Chapter II),

$$U = \sum_i \frac{1}{2} k_l (l_i - l_0)^2 + \sum_i \frac{1}{2} k_\theta (\theta_i - \theta_0)^2 + \sum_i \frac{V_n}{2} [1 + \cos(\bar{n}\varphi_i + \delta)] \qquad \text{Equation 4.1}$$
$$+ \sum_{i,l} \varepsilon_{i,l} \left(\left(\frac{\sigma_{i,l}}{r_{i,l}} \right)^{12} - 2 \left(\frac{\sigma_{i,l}}{r_{i,l}} \right)^{6} \right)$$

where $r_{i,l}$ is the distance between residues, or in this case carbons, separated by two or more residues along the central chain, $k_l = 310.0\left(\frac{kcal}{Ang^2}\right)$, $k_\theta = 40.0\left(\frac{kcal}{rad^2}\right)$, $V_n = 1.3\,(kcal)$, $\varepsilon_{i,l} = 0.181\,(kcal)$, $\sigma_{i,l} = 4.0\,(Ang)$, $l_0 = 1.526\,(Ang)$, and $\theta_0 = 109.47°$. Notice that l_0 and θ_0 represent the preferred bond angles and bond lengths, respectively, and n is set to three, providing three preferred torsion angles at 60°, 180°, and 300°, while δ=0°. Since all bonds between hydrocarbons can be considered identical, only one value of the $\varepsilon_{i,l}$ and $\sigma_{i,l}$ is required.

Even while great simplifications have been made (modeling the molecule as a chain of beads, assuming all residues to be identical, and using a small example, such as heptane), the conformation problem remains very difficult. Since torsion angles have three preferred positions (60°,180°,300°), 3^{n-3}, or 81, local minima are expected. (Recall that three of the torsion angles are fixed.) Tests on the potential function support this estimate; by starting from 1000 random starting points, 77 local minima were discovered. However, merely selecting preferred angles does not guarantee that the initial conformer is also a local minimum. (Notice in Table 4.1 that the local minimum found by starting at (-60,60,-

Table 4.1 Selected Minima of Heptane

U(ϕ)	Initial Position, $\phi^{(0)}$				Minimized Position, ϕ^*			
-0.339727	180	180	180	180	180.000000	180.000000	180.000000	180.000000
-0.215609	-60	60	-60	60	-62.657878	179.600918	-179.857154	179.956953
-0.215524	-60	180	180	180	-62.706316	179.627701	180.119720	180.000000

Table 4.1 Selected Minima of Heptane

$U(\phi)$	Initial Position, $\phi^{(0)}$				Minimized Position, ϕ^*			
-0.104118	60	60	-60	60	62.550498	180.191468	-179.798849	62.592457
2.097992	180	-60	60	60	179.378780	-79.278332	79.236470	62.423103
4.384072	180	-60	60	-60	178.741629	-93.723491	69.284953	-94.158124

60,60) results in the same local minimum as the point with initial torsion angles (-60,180,180,180).) The global minimum for heptane corresponds to the point with all torsion angles equal to 180 (see Table 4.1 and Figure 4.2). (For a complete list of the local

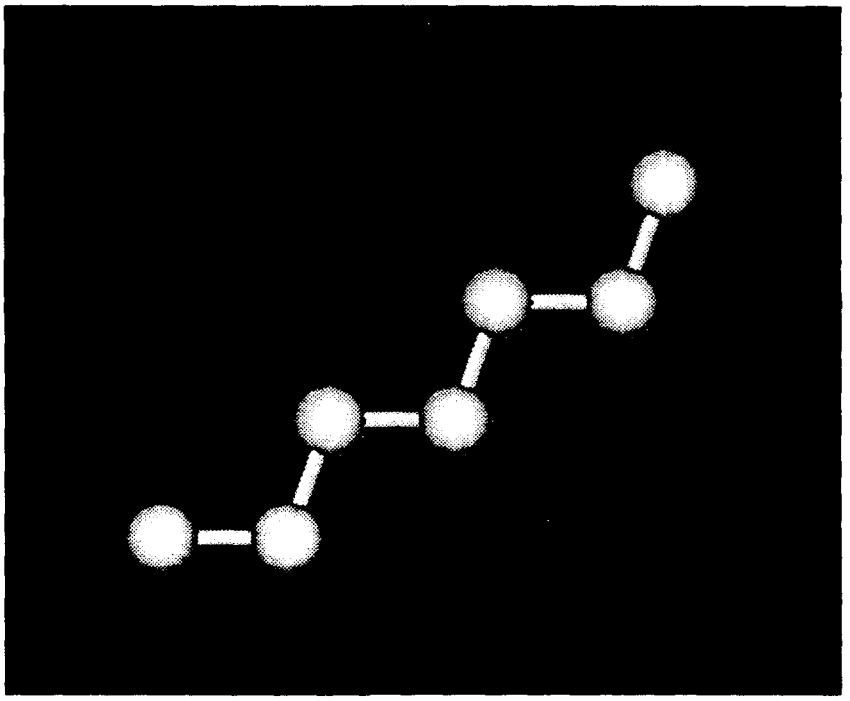


Figure 4.2 Global Minimum Conformation, ϕ^*, for Heptane

minima for heptane found from the preferred torsion angles, see Appendix A.)

While it may seem that 77 local minima is relatively small, larger *n*-chain hydrocarbons produce an exponentially increasing number of local minima. A molecule with as few as 12 beads has 19,683 possible preferred angle configurations, with most of these

likely to produce a local minimum. It can clearly be seen that finding the global minimum for any problem approaching the size of the protein folding problem (lengths of 20 or more), is intractable with exhaustive methods. The next few chapters discuss methods which attempt to find the global solution without resorting to methods which grow exponentially intractable with problem size.

V. Convex Global Underestimator

A. A Global Underestimator

One possibility for aiding the search for the global minimum of the molecule's potential energy function is to use a global underestimator to localize the search in the region of the global minimum. This new method is designed to fit all known local minima with a convex function which underestimates all of them, but which differs from them by the minimum possible amount in the discrete L_1 norm. The minimum of this underestimator is used to predict the global minimum for the function, allowing a localized conformer search to be performed based on the predicted minimum. A new set of conformers generated by the localized search can then serve as a basis for another quadratic underestimation. After several repetitions, the global minimum can be found with reasonable assurance.

The use of an underestimating function allows the translation of a ***very*** complex function into a simple underestimator. If the underestimator is well-suited to the problem (provides accurate predictions for the global minimum), immense savings in time can be accomplished. The presence of quadratic terms in the length and bond angle portions of the molecular energy function (See Equation 2.1.) indicate the use of a convex quadratic may provide a suitable approximation for this problem.

B. Theory

This method is presented in terms of a differentiable function $F(\phi)$, where $\phi \in \mathbf{R}^{\tau}$ (τ represents the number of torsion angles), and where $F(\phi)$ has many local minima. Since the bond lengths and bond angle terms in the potential function carry severe quadratic

penalties, they can be assumed to be fixed for the purposes of the global underestimator. Therefore, $F(\phi)$ is a vector of torsion angles of length τ, where $\tau = n - 3$ and n is the number of residues. (Recall that the first three are fixed by definition.) To begin the iterative process, a set of $k \geq 2\tau+1$ distinct local minima are computed. This can be done with relative ease by using an efficient unconstrained minimizer, starting with a large set of points chosen at random in the hypercube $H\phi$, which is assumed to enclose the torsion angle space.

Assuming that $k \geq 2\tau+1$ local minima $\phi^{(j)}$, for j=1,...,k, have been computed, a convex quadratic underestimator function $\Psi(\phi)$ is now fitted to these local minima so that it underestimates all the local minima and normally interpolates $F(\phi^{(j)})$ at $2\tau+1$ points. This is accomplished by determining the coefficients in the function $\Psi(\phi)$ so that:

$$\delta_j = F(\phi^{(j)}) - \Psi(\phi^{(j)}) \geq 0 \qquad \text{Equation 5.1}$$

for j=1,...,k, and where $\sum_{j=1}^{k} \delta_j$ is minimized. That is, the difference between $F(\phi)$ and $\Psi(\phi)$ is minimized in the discrete L_1 norm over the set of k local minima $\phi^{(j)}$, j=1,...,k. The underestimating function $\Psi(\phi)$ is given by:

$$\Psi(\phi) = c_0 + \sum_{i=1}^{\tau} \left(c_i \phi_i + \frac{1}{2} d_i \phi_i^2 \right). \qquad \text{Equation 5.2}$$

Note that c_i and d_i appear linearly in the constraints (Equation 5.1) for each local minimum $\phi^{(j)}$. Convexity of this quadratic function is guaranteed by requiring that $d_i \geq 0$ for $i=1,...,\tau$. Other linear combinations of convex functions could also be used, but this quadratic function is the simplest.

C. Minimizing the Underestimating Function

The unknown coefficients c_i, i=0,...,τ, and d_i, i=1,...,τ, can be determined by a linear program which may be considered to be in the dual form. For reasons of efficiency, the equivalent primal of this problem is actually solved, as described below. The solution to this primal linear program provides an optimal dual vector, which immediately gives the underestimating function coefficients c_i and d_i. Since the convex quadratic function $\Psi(\phi)$ gives a global approximation to the local minima of $F(\phi)$, then its easily computed global minimum function value Ψ_{min} is a good candidate for an approximation to the global minimum of the correct energy function $F(\phi)$.

An efficient linear programming formulation and solution satisfying Equation 5.1 and Equation 5.2 will now be summarized. Let $f^{(j)} = F(\phi^{(j)})$, for j=1,...,k, and let $f \in \mathbf{R}^k$ be the vector with elements $f^{(j)}$. Also let $\omega^{(j)} \in \mathbf{R}^\tau$ be the vector with elements $1/2(\phi^{(j)}_i)^2$, i=1,...,τ, and $e_k \in \mathbf{R}^k$ be the vector of ones. Now define the following two matrices $\Phi \in \mathbf{R}^{(\tau+1)\times k}$ and $\Omega \in \mathbf{R}^{\tau\times k}$:

$$\Phi = \begin{bmatrix} e_k^T \\ \phi^{(1)}\,\phi^{(2)}\ldots\phi^{(k)} \end{bmatrix}, \quad \Omega = \left[\omega^{(1)}\,\omega^{(2)}\ldots\omega^{(k)}\right]. \qquad \text{Equation 5.3}$$

Finally, let $c \in \mathbf{R}^{\tau+1}$, $d \in \mathbf{R}^\tau$, and $\delta \in \mathbf{R}^k$ be the vectors with elements c_i, d_i, and δ_i, respectively. Then (Equation 5.1) and (Equation 5.2) can be restated as the linear program (with variables c, d, and δ):

$$\underset{c,\,d,\,\delta}{\text{minimize}} \quad e_k^T\delta$$

$$\text{subject to} \begin{bmatrix} \Phi^T & \Omega^T & 0 \\ -\Phi^T & -\Omega^T & -I_k \\ 0 & -I_\tau & 0 \end{bmatrix} \begin{bmatrix} c \\ d \\ \delta \end{bmatrix} \leq \begin{bmatrix} f \\ -f \\ 0 \end{bmatrix}. \qquad \text{Equation 5.4}$$

It is easy to see that c = d = 0 and $\delta = f \geq 0$ is always feasible for (Equation 5.4), and since $e_k^T \delta \geq 0$, the linear program always has an optimal solution. Furthermore, since the matrix in (Equation 5.4) has more rows than columns ($2k+\tau$ rows and $k+2\tau+1$ columns, where $k \geq \tau+1$), it is computationally more efficient to consider it as a dual problem, and to solve the equivalent primal. After some simple transformations, this primal problem reduces to:

$$\underset{y_1, y_2}{\text{minimize}} \quad f^T y_1 - f^T e_k$$

$$\text{subject to} \begin{bmatrix} \Phi & 0 \\ \Omega & -I_\tau \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \Phi e_k \\ \Omega e_k \end{bmatrix}, \quad y_1, y_2 \geq 0 \qquad \text{Equation 5.5}$$

which has only $2\tau+1$ rows and $k+\tau \geq 3\tau+1$ columns, and the obvious initial feasible solution $y_1 = e_k$ and $y_2 = 0$. Its optimal solution gives the values of c, d, and δ via the dual vectors, and also determines which values of $f^{(j)}$ are interpolated by the potential function $\Psi(\phi)$. That is, the basic columns in the optimal solution to (Equation 5.5) correspond to the conformations $\phi^{(j)}$ for which $f^{(j)} = \Psi(\phi^{(j)})$.

Note that once an optimal solution to (Equation 5.5) has been obtained, the addition of new local minima is very easy. It is done by simply adding new columns to Φ and Ω, and therefore to the constraint matrix in (Equation 5.5). The number of primal rows remains fixed at $2\tau+1$, independent of the number k of local minima.

D. Iteration - Approaching the Global Minimum

The convex quadratic underestimating function $\Psi(\phi)$ determined by the values $c \in \mathbf{R}^{\tau+1}$ and $d \in \mathbf{R}^{\tau}$ now provides a global approximation to the local minima of $F(\phi)$, and its easily computed global minimum point ϕ_{min} is given by $(\phi_{min})_i = -c_i/d_i$, $i=1,...,\tau$, with corresponding function value Ψ_{min} given by $\Psi_{min} = c_0 - \sum_{i=1}^{\tau} c_i^2/(2d_i)$. The value Ψ_{min} is a good candidate for an approximation to the global minimum of the correct energy function $F(\phi)$, and therefore ϕ_{min} can be used as an initial starting point around which additional configurations (i.e. local minima) should be generated. These local minima are added to the constraint matrix in (Equation 5.5) and the process is repeated. Before each iteration of this process, it may be advantageous to reduce the volume of the hypercube $H\phi$ over which the new configurations are produced so that a tighter fit of $\Psi(\phi)$ to the local minima "near" ϕ_{min} may be produced.

The rate and method by which the hypercube size is decreased and the number of additional local minima computed at each iteration will be determined by computational testing. The selection of the hypercube size can be done computationally or with the aid of an expert. But clearly the method depends most heavily on computing local minima quickly and on solving the resulting linear program efficiently to determine the approximating function $\Psi(\phi)$ over the current hypercube.

If E_c is a cutoff energy, then one means for decreasing the size of the hypercube $H\phi$ at any step is to let $H\phi = \{\phi: \Psi(\phi) \leq E_c\}$. To get the bounds of $H\phi$, consider $\Psi(\phi) \leq E_c$ where $\Psi(\phi)$ satisfies (Equation 5.2). Then limiting ϕ_i requires that:

$$\sum_{i=1}^{\tau} \left(c_i\phi_i + \frac{1}{2} d_i\phi_i^2 \right) \leq E_c - c_0. \qquad \text{Equation 5.6}$$

As before, the minimum value of $\Psi(\phi)$ is attained when $\phi_i = -c_i/d_i$, $i=1,...,\tau$. Assigning this minimum value to each ϕ_i, except ϕ_k, then results in:

$$c_k\phi_k + \frac{1}{2}d_k\phi_k^2 \le E_c - c_0 + \frac{1}{2}\sum_{i \ne k} c_i^2/d_i \equiv \beta_k \,. \qquad \text{Equation 5.7}$$

The lower and upper bounds on ϕ_k, $k=1,...,\tau$, are given by the roots of the quadratic equation

$$c_k\phi_k + \frac{1}{2}d_k\phi_k^2 = \beta_k \,. \qquad \text{Equation 5.8}$$

Hence, these bounds can be used to define the new hypercube $H\phi$ in which to generate new configurations.

E. Local Minimization

The final predicted point can be used as the initial point for a local minimization procedure. Since the point predicted by the underestimator is expected to be very close to the global minimum, the conformation found at the nearest local minimum should correspond to the global minimum.

An accurate and efficient local minimization procedure is of great importance to the establishment of an accurate underestimator as well. The effectiveness of the global underestimating procedure relies heavily upon the ability to produce very many local minima. Since this minimization is performed a large number of times and each minimum must reflect an accurate conformer, the development an efficient algorithm is extremely important.

VI. Finding Local Minima

A. Twice Differentiable Unconstrained Optimization

Since the potential energy function is smooth (twice-continuously differentiable), any method which computes an n-dimensional unconstrained minimum can be used. The methods to be examined satisfy the basic form (where ϕ_k is the current estimate of ϕ^*, the global minimizer conformation):

1. Obtain some initial conformation φ_0 and set $k \leftarrow 0$.
2. [Test for convergence.] If the conditions for convergence are satisfied, the algorithm terminates with ϕ_k as the solution.
3. [Compute a search direction.] Compute a non-zero n-vector p_k, the direction of search.
4. [Compute a step length.] Compute a positive scalar α_k, the step length, which satisfies the descent condition, $F(\phi_k + \alpha_k p_k) < F(\phi_k)$.
5. [Update the estimate of the minimum.] Set $\phi_{k+1} \leftarrow \phi_k + \alpha_k p_k$, $k \leftarrow k+1$, and go back to step 2.

To satisfy the descent condition (that for a positive scalar, $F(\phi_k + \alpha_k p_k) < F(\phi_k)$), p_k and α_k must have certain properties. One way of guaranteeing that F can be reduced at the kth iteration is to require that p_k satisfies

$$g_k^T p_k < 0 \qquad \text{Equation 6.1}$$

where $g_k = \nabla F(\phi_k)$ is the gradient of the function, F, at point ϕ_k. When this condition is satisfied, p_k is known as a descent direction. The next three sections will describe three

related, but different, methods for computing the local minima of F(ϕ).

B. Steepest Descent

Perhaps the greatest impact on the effectiveness of a local minimization algorithm is the selection of a "good" search direction, p_k. Even in as few as two dimensions there are an infinite number of ways to compute the descent direction. The importance of this decision led to the naming of local minimizers according to the procedure by which it computes the search direction.

Since a linear approximation to the function F(ϕ) can be presented as a Taylor series expansion about x_k (where g_k denotes $g(\phi_k) = \nabla F(\phi_k)$),

$$F(\phi_k + p) \approx F(\phi_k) + g_k^T p \qquad \text{Equation 6.2}$$

it would appear that a large reduction in F would occur when $g_k^T p$ is large and negative. However, limits must be placed on p. Otherwise, for any $\bar{p}$ such that $g_k^T \bar{p} < 0$, p could simply be chosen as an arbitrarily large multiple of $\bar{p}$. The aim is to chose p, so that among all normalized vectors, $g_k^T p$ is a minimum.Therefore, given some norm, $\|\bullet\|$, p_k is the solution of the minimization problem:

$$\underset{\substack{p \in \Re^n \\ \|p\| = 1}}{\text{minimize}} \left(g_k^T p \right). \qquad \text{Equation 6.3}$$

If the two-norm is used for the calculation of $\|p\|$, the solution to Equation 6.3 is just the negative gradient,

$$p_k = -g(\phi_k). \qquad \text{Equation 6.4}$$

This solution is known as the direction of steepest descent. Clearly, as long as the gradient does not equal zero, a descent direction is guaranteed.

Unfortunately, while a steepest descent algorithm is relatively simple and is guaranteed to converge (reach a minimum), it is not very efficient. When faced with a function of quadratic or greater complexity, the steepest descent algorithm may require several hundred iterations to make very little progress towards the solution. The function $F(\phi)$ presents just such a problem. Hence more efficient methods must be explored (Gill et al. 1981).

C. Newton's Method

The primary weakness of the steepest descent method is its attempt to represent a complex function using a linear model involving only the gradient. The use of a quadratic model retains much of the simplicity found in the steepest descent procedure, yet can yield greater success and efficiency in practice.

This model, based on the first and second derivatives of the function F, is known as Newton's Method. The function F is approximated using the first three terms of the Taylor-series expansion:

$$F(x_k + p) \approx F_k + g_k^T p + \frac{1}{2} p^T G_k p \qquad \text{Equation 6.5}$$

where G_k denotes the Hessian, $\nabla^2 F(\phi)$ at the point ϕ_k. It is helpful to formulate the algorithm in terms of p (the step to the minimum), rather than the predicted minimum itself. A minimum of the right hand side of Equation 6.5 will be achieved at the minimum of the quadratic:

$$\Phi(p) = g_k^T p + \frac{1}{2} p^T G_k p . \qquad \text{Equation 6.6}$$

The function Φ has a stationary point only if there exists a point where its gradient vector

vanishes (becomes zero). Therefore, for any minimum, the gradient of Φ must be equal to zero; that is, $\nabla\Phi(p) = G_k p + g_k = 0$. A minimization algorithm in which p_k is defined by:

$$p_k = -G_k^{-1} g_k \qquad \text{Equation 6.7}$$

is termed Newton's method, and the direction, p_k, a Newton direction.

If G_k is positive definite and F(ϕ) is itself quadratic, only one iteration is required to reach the minimum of the model function from *any* starting point. Therefore, good convergence from the Newton's method is expected when the quadratic model is accurate. For a general function (F) Newton's method converges *quadratically* to ϕ^* if the starting point (ϕ_0) is sufficiently close to ϕ^*, the Hessian matrix is positive definite at ϕ^*, and the step lengths (α_k) converge to unity.

The local convergence properties of Newton's method make it an exceptionally attractive algorithm for unconstrained optimization. The availability of the second derivative also allows the verification that sufficient conditions for optimality exist. (Gill et al. 1981) While Newton's method can be an extremely effective optimization method, it requires the often very difficult computation of second derivatives.

D. Quasi-Newton Methods

Indeed, F(ϕ) presents a very difficult case since not only are the second derivatives difficult to compute, so are the first! Ideally, advantage of the curvature information (the Hessian matrix) could be taken without having to compute it directly. Just such an attempt is made in quasi-Newton methods. These algorithms are based on the idea of "building up" curvature information as the iterations of a descent method proceed. Thus, an approx-

imation to the curvature of a non-linear function can be computed without explicitly forming the Hessian matrix.

Quasi-Newton methods are very similar to Newton's method, except that $\mathbf{G}_k^{-1}$ is *approximated* by a symmetric, positive definite matrix, $\mathbf{H}_k$, which is corrected or updated from iteration to iteration. The basic structure for the *k*th iteration is:

1. Set the search direction, $\mathbf{s}_k = -\mathbf{H}_k\mathbf{g}_k$.
2. Compute α_k using line search along $\mathbf{s}_k$ giving $\phi_{k+1} = \phi_k + \alpha_k\mathbf{s}_k$.
3. Update $\mathbf{H}_k$ giving $\mathbf{H}_{k+1}$.

The initial matrix, $\mathbf{H}_1$ can be any positive definite matrix, although in the absence of better estimates, $\mathbf{H}_1=\mathbf{I}$ is often made.

The update of the Hessian matrix is generally accomplished through the use of an *updating formula.* Many formulas have been established, each of which attempts to generate a Hessian, $\mathbf{H}_{k+1}$, from the previous Hessian $\mathbf{H}_k$ and preserve positive definiteness. Through the use of second derivative information gained on the *k*th iteration, repeated updating can change the arbitrary matrix, $\mathbf{H}_1$, into a close approximation, $\mathbf{H}_k$, to $\mathbf{G}_k^{-1}$.

One possibility for updating the Hessian was developed by Broyden, Fletcher, Goldfarb, and Shanno, and is known as the BFGS formula (Fletcher 1987). If the differences,:

$$\delta_k = \alpha_k s_k = \phi_{k+1} - \phi_k \qquad \text{Equation 6.8}$$

$$\gamma_k = g_{k+1} - g_k \qquad \text{Equation 6.9}$$

are defined, then the BFGS formula can be stated as:

$$\mathbf{H}_{k+1} = \mathbf{H} + \left(1 + \frac{\gamma^T \mathbf{H} \gamma}{\delta^T \gamma}\right)\frac{\delta\delta^T}{\delta^T\gamma} - \left(\frac{\delta\gamma^T\mathbf{H} + \mathbf{H}\gamma\delta^T}{\delta^T\gamma}\right). \qquad \text{Equation 6.10}$$

Note that the subscript, k, has been suppressed on the right hand side. While many other formulas have been developed, none have shown any significant improvement over the BFGS method.

Quasi-Newton methods provide several distinct advantages over a Newton's method optimization:

1. Only first derivatives are required.
2. Positive definiteness of $\mathbf{H}_k$ will guarantee the descent property.
3. Only $O(n^2)$ multiplications per iteration are required. ($O(n^3)$ are required for Newton's method.)

E. Finite Difference Computations

Unfortunately, computation of even the first derivative can be a source of great difficulty, as is the case with the potential function given by due to the mixture of cartesian and chemical representations. However, the ability to expand smooth functions in Taylor series allows the values of derivatives to be approximated. This is, if h_j is the finite difference interval (the spacing over which the approximation is to be made) associated with the j-th component of ϕ and e_j is the vector with unity in the j-th position and zeros elsewhere, then:

$$F(\phi + h_j e_j) = F(\phi) + h_j g^T e_j + \frac{1}{2} h_j^2 e_j^T G(\phi + \rho_1 h_j e_j) e_j \qquad \text{Equation 6.11}$$

where $0 \le \rho_1 \le 1$. Since the terms $g^T e_j$ and $e_j^T G(\phi + \rho_1 h_j e_j) e_j$ are the j-th element of g, and the j-th diagonal element of $G(\phi + \rho_1 h_j e_j)$, respectively, then:

$$g_j = \frac{1}{h_j}\left(F(\phi + h_j e_j) - F(\phi)\right) + O(h_j)\,. \qquad \text{Equation 6.12}$$

Equation 6.12 describes the *forward-difference interval* expression in the direction e_j. The error developed in approximating g_j by the expression to the left of Equation 6.12 is due to neglecting the term, $\frac{1}{2}h_j^2 e_j^T G(\phi + \rho_1 h_j e_j)\, e_j$. This inaccuracy is known as the *truncation error*, since it arises from the truncation of the Taylor series. In general, ρ_1 is unknown and therefore it is possible only to compute or estimate an upper bound on the truncation error. Similarly, a *backward-difference* expression can be developed by expanding the Taylor series about the point $(\phi - h_j e_j)$ in order to get:

$$g_j = \frac{1}{h_j}\left(F(\phi) - F(\phi - h_j e_j)\right) + O(h_j)\,. \qquad \text{Equation 6.13}$$

If the Taylor series expansions of F(ϕ) using both the forward and backward difference equations are carried one term further, the result is:

$$F(\phi + h_j e_j) = F(\phi) + h_j g_j^T e_j + \frac{1}{2}h_j^2 e_j^T G_j e_j + O\left(h_j^3\right) \qquad \text{Equation 6.14}$$

and:

$$F(\phi - h_j e_j) = F(\phi) - h_j g_j^T e_j + \frac{1}{2}h_j^2 e_j^T G_j e_j + O\left(h_j^3\right). \qquad \text{Equation 6.15}$$

Subtracting and dividing by h_j, one obtains:

$$g_j(\phi) = \frac{1}{2h_j}\left[F(\phi + h_j e_j) - F(\phi - j_j e_j)\right] + O\left(h_j^2\right) \qquad \text{Equation 6.16}$$

since the terms containing G_j cancel. The right hand side of this equation is known as a *central-difference approximation* to $g_j(\phi)$. This formulation provides a truncation error of

the second order, involving the third derivative of F in the region $[\phi\text{-}h_j e_j, \phi\text{+}h_j e_j]$. However, two function evaluations (in addition to $F(\phi)$) are required to approximate $g_j(\phi)$, as compared to one evaluation for Equation 6.12 and Equation 6.13 (the forward and backward-difference approximations).

Through the use of the central difference approximation (Equation 6.16), the gradient of the function can be calculated without explicit knowledge of the derivative. The error produced in the finite difference approximation should be small as long as h_j is chosen to be suitably small (Gill et al. 1981).

F. Step Length (Line Search Calculations)

Once the search direction, s, has been established, the distance to travel along this path, α, must be found. Any line search method will suffice for this calculation. Since the potential function is of unknown complexity, a simple algorithm is chosen:

1. Let $f_0 = F(\phi)$.
2. Set α_0 to a *very* small initial value ($\alpha \leftarrow \varepsilon > 0$) and $i \leftarrow 0$.
3. Compute a new function value, $f_{i+1} = F(\phi + \alpha_i s)$.
4. If $f_{i+1} < f_i$, set $\alpha_{i+1} = 2\alpha_i$, $i \leftarrow i + 1$, and return to step 3.
5. Otherwise, set $\alpha = \alpha_i$ and stop.

VII. Underestimating Function- Trials and Results

A. Global Minimum Conformation for Heptane

A solution for the 7 bead hydrocarbon heptane was obtained in 68 seconds of CPU time on a SUN Sparcstation 2. While the improvements in computation time are small for this problem (Enumeration required only 79 seconds to test all local minima.), many of the characteristics of the method can be observed. Table 7.1 summarizes the minimization of the potential function for heptane. (See Appendix C for a complete listing of the program output.) Notice that the first iteration reports the current "best" (corresponding to the lowest potential energy value) conformation to be approximately the set of torsion angles, (180°,180°,300°,180°). Recall that the bond lengths and bond angles are considered fixed

Table 7.1 Run Summary for Heptane

	Initial Position, $\phi^{(0)}$	Randomize over Radius	New Global Minimum (25 random points)	Predicted Global Minimum	Randomize over Radius	New Global Minimum (20 random points)	Predicted Global Minimum	Randomize over Radius	New Global Minimum (10 random points)	Predicted Global Minimum
φ_1	20	180	180.12	195.45	20	180.01	NC	4	180.02	NC
φ_2	330	180	179.53	187.96	15	179.99	NC	4	179.99	179.88
φ_3	278	180	297.50	163.06	140	180.06	180.14	5	180.12	NC
φ_4	-60	180	179.63	195.98	20	179.97	179.88	5	179.91	NC
$F(\phi)$			-.2967			-.3398			-.3398	
									ϕ^*	

for the purposes of the global underestimator. However, the global underestimator predicts a conformation close to (180°, 180°, 180°, 180°). After the second iteration, concentrated around the predicted point, the new results report that a better function value can be realized at this conformation. The underestimator and the current global minimum now reside at almost identical points. The third iteration confirms (180°, 180°, 180°, 180°) as the global minimum. This result has been shown through enumeration to be the correct global minimum conformation for the potential energy function.

B. Prediction for Hydrocarbons of Arbitrary Length

As problem size increases, the number of local minima increases exponentially ($O(3^n)$), as can be seen in Figure 7.2. With exponential increases in the number of local

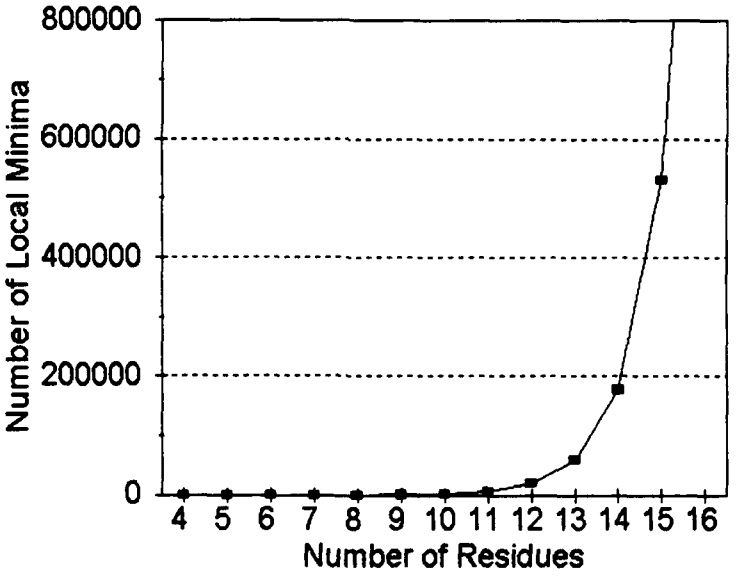


Figure 7.2 Number of Residues vs. Number of Local Minima

minima, exhaustive searches for the global minimum quickly become computationally intractable. By comparison, the underestimator method produces a final prediction in $O(n^4)$ time.

The main contributors to the underestimator's $O(n^4)$ rate are the function evalua-

tions and conversions. Every function evaluation incurs an $O(n^2)$ penalty due to the conversions and distance calculations required by the Lennard-Jones term. The non-linear program which computes the local minima averages $O(n)$ iterations to reach a local minimum (Figure 7.3). The $O(n)$ iterations performed in the local minimization procedure,

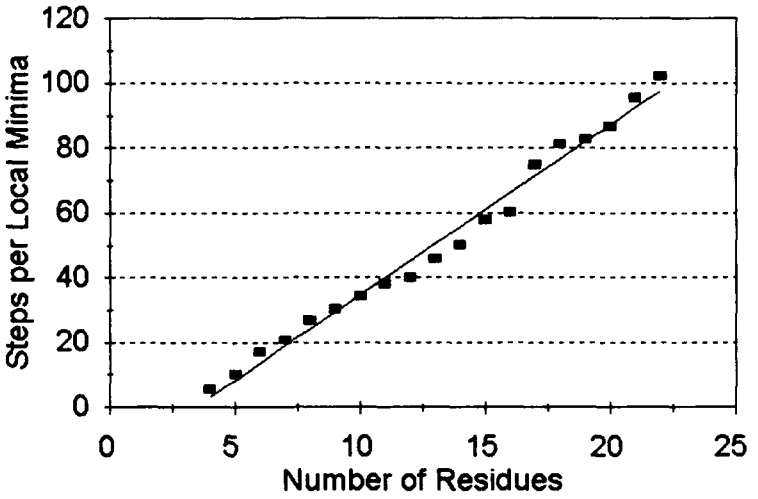


Figure 7.3 Number of Iterations Required for Local Minimization

combined with the $O(n^2)$ time for function evaluation, indicate that the total time required per local minimization increases at a rate no greater than $O(n^3)$ (See Figure 7.4.). In fact, the time required to perform a single local minimization as a function of the number of beads, n, can be approximated (using a least squares method) by the function $f(n) = 0.003n^3 - 0.68n^2 + 5.46n - 14.28$ (as seen by the curve in Figure 7.4).

The amount of time for exhaustive techniques is driven by the number of local minimizations required to test each case (531,441 minimizations in the 15 bead problem), and hence, to test every possible local minimum conformation for an n-chain hydrocarbon of length 15 would take over 185 days on a Sun workstation! In contrast to enumeration, the underestimator requires a much less drastic increase in the number of local minima calculations. Since the number of iterations of the underestimator needed to find a global

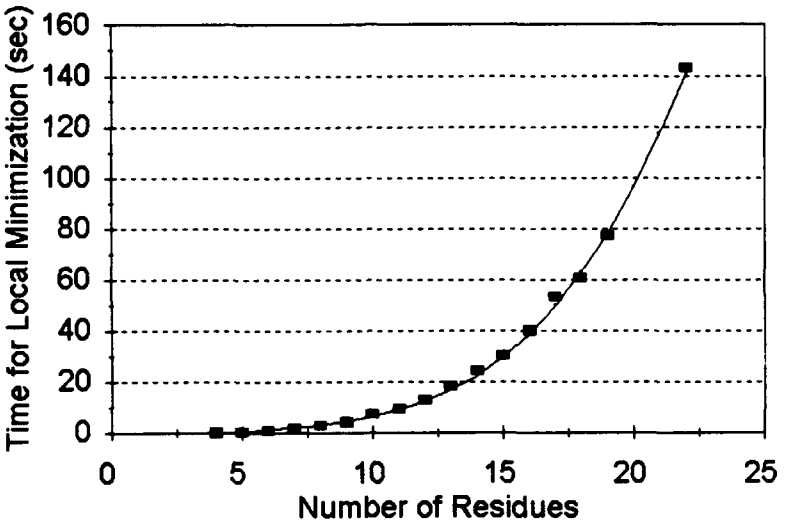


Figure 7.4 Time for Local Minimization per Number of Residues

minimum remains close to constant (O(1)), and each step requires O(n) local minima to form the underestimator, a linear increase in the number of minimizations is expected. This effect can be seen in Figure 7.5.

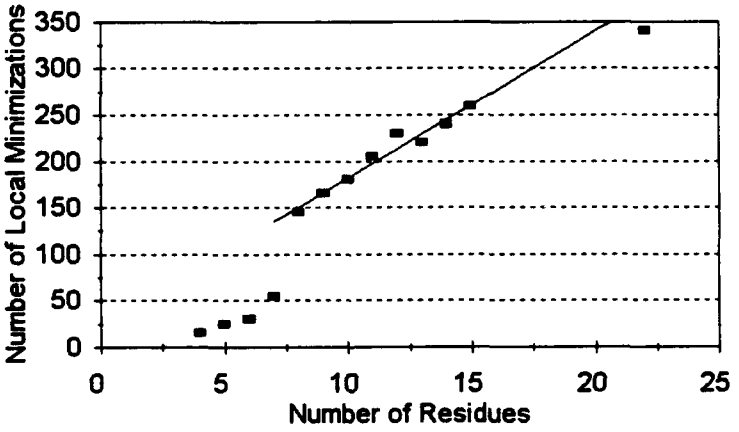


Figure 7.5 Number of Local Minima for Complete Global Minimum Prediction

The effect of increases in problem size on total time for global minimum prediction can be seen in Figure 7.6. Note that this graph depicts CPU time and does not include the time required to enter values between iterations. The curve seen in Figure 7.6 repre-

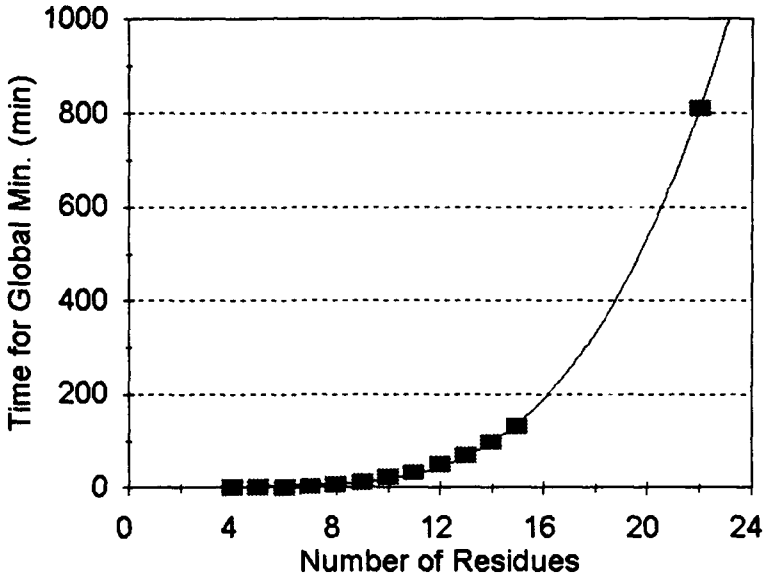


Figure 7.6 Total CPU Time for Global Minimum Prediction

sents the least squares fit, $0.01n^4 - 0.29n^3 + 3.85n^2 - 22.26n + 44.13$. Thus the final prediction can be provided in no worse than $O(n^4)$ time. While the prediction times for the underestimator are clearly significantly better than those of an exhaustive search, the $O(n^4)$ rate of increase in time as problem size grows makes it unlikely that molecules substantially larger than 22 beads will be solved. In fact, the underestimating method enabled predictions to be made for hydrocarbons of length 4 through 15 and 22. (See Appendices B, C, D, and E for selected underestimator runs.)

The 22 bead problem took over 16 hours for the final prediction (Figure 7.7). This prediction was made with 4 iterations of the global underestimator (See Appendix E for complete output from the 22 bead run.), and 143 seconds were required for each local minimization. Further increases in problem size will face even larger increases in execution time.

C. Applicability to the Protein Folding Problem

While the molecular conformation problem is a greatly simplified version of the

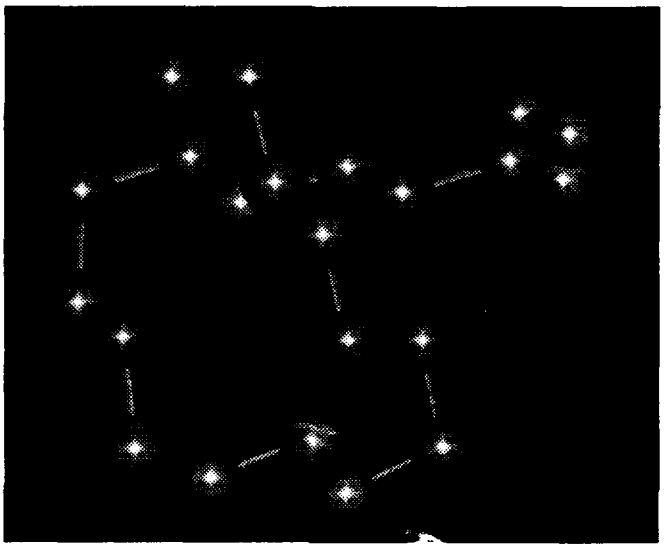


Figure 7.7 Global Energy Minimum for a 22 bead hydrocarbon

protein folding problem, the basic "unified atom" model used is still applicable to both. The only substantial change is that the potential energy function should be appropriate to the protein structure. The new potential energy function would be required to model interactions between all combinations of the 20 amino acids, as well as take into account any additional potential energy contributions caused by amino acid interactions. Notice, however, that this model completely ignores any intra-peptide interactions; that is, the model used does not consider any secondary structures such as alpha helixes and beta strands. A much more complicated model would be required to capture these interactions. Nevertheless, the formulation of the convex quadratic global underestimator and the local minimization procedures based on the model, presented in Chapter II, will remain applicable to the problem.

VIII. Quadratic Assignment Formulation

A. Lattice Restrictions

Attempts to solve the protein folding problem in continuous space confront the daunting task of finding the optimal solution from among an infinite number of conformations. Lattice methods attempt to avoid this by restricting each amino acid to a vertex, or site, on a fixed lattice. By limiting the positions of the residues, the potential energy minimization can be formulated in a manner which facilitates the optimization. This section describes a method which involves a quadratic assignment formulation for the protein folding problem using the lattice model. The quadratic assignment problem can then be transformed into a continuous concave quadratic global minimization problem. The global solution to this concave minimization problem can then be used as a starting point for a "relaxed" continuous minimization problem. The result this minimization should provide a global, or near global, minimum of the potential energy function, and hence a prediction of the native, or folded state, of the molecule (Phillips and Rosen 1994).

B. Quadratic Assignment Formulation

To formulate a discrete approximation to the molecular conformation problem, the original continuous problem in 3-dimensional space is approximated by a suitable 3-dimensional lattice with N sites, $N \geq n$ (where n is the number of amino acids). If s_j represents lattice site j, for $j= 1,...,N$, then a total of nN zero-one variables x_{ij} are sufficient to completely determine the assignment of the beads a_i to the lattice sites s_j. Two constraints are established:

1. Each bead must occupy exactly one lattice site.

$$\sum_{j=1}^{N} x_{ij} = 1, \qquad i = 1, \ldots, n$$

Equation 8.1

2. At most one bead occupies each lattice site, s_j.

$$\sum_{j=1}^{n} x_{ij} \leq 1, \qquad i = 1, \ldots, N$$

Equation 8.2

The objective function can be written in two portions, a linear and quadratic term. The linear term:

$$\sum_{i=1}^{n} \sum_{j=1}^{N} d_{ij} x_{ij}$$

Equation 8.3

represents the direct contribution, d_{ij}, to the potential energy when the bead a_i is assigned to lattice site s_j. Direct contributions consist of the interaction of the residue, a_i, and the environment (hydrophobic beads might prefer to be in the interior of the folded chain and hence in the center of the lattice structure). The quadratic term:

$$\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{N} \sum_{k=1}^{n} \sum_{l=1}^{N} p_{ijkl} x_{ij} x_{kl}$$

Equation 8.4

represents the pairwise contribution p_{ijkl} to the total potential energy when the bead a_i is assigned to lattice site s_j and bead a_k is assigned to lattice site s_l. One possibility for this penalty is, as in the continuous case, the Lennard-Jones pairwise potential. Using Lennard-Jones, the term p_{ijkl} depends only on a_i, a_k, and $\|s_j - s_l\|$. Therefore, it is convenient to write the total potential energy function in the form:

$$E(x) = c^T x + \frac{1}{2} x^T Q x$$

Equation 8.5

where $x \in \mathbf{R}^{nN}$ denotes the zero-one vector with elements x_{ij}, $c \in \mathbf{R}^{nN}$ denotes the vector with elements d_{ij}, and $Q \in \mathbf{R}^{(nN) \times (nN)}$ is a real symmetric (typically indefinite) matrix with elements p_{ijkl}. Thus, the 3-dimensional minimum energy conformation of the lattice restricted model is given by the solution to the *quadratic assignment problem*:

$$\min_{x \in P} E(x) \qquad \text{Equation 8.6}$$

where P consists of the constraints (Equation 8.1) and (Equation 8.2), and the integer restrictions, $x_{ij} \in \{0,1\}$ for $i = 1,...,n$ and $j = 1,...,N$.

The usefulness of this approach is very heavily dependent on the selection of the lattice structure. Obviously, determining the perfect lattice is as difficult as the protein folding problem itself. Great care must be taken to insure the lattice is fine enough to closely approximate possible solutions in continuous space, and coarse enough so that the problem does not become computationally intractable. The lattice structure can take many forms: spherical, rectangular, random, or specially developed (Hinds and Levitt 1992). That the lattice might not allow for the global minimum, or any stable conformation at all, is not important. The continuous minimization stage will relax the "non-stable" conformer found in the lattice restricted phase to a true energy minimum.

C. Concave Quadratic Global Minimization Formulation

The discrete quadratic assignment problem (Equation 8.6) can be shown to be equivalent to the continuous minimization of a strictly concave quadratic function over a polytope. If λ_{max} is the maximum eigenvalue of the matrix $Q = \mathbf{R}^{(nN) x (nN)}$, and $\mu = 1 + \lambda_{max}$, then since $x_{ij}^2 = x_{ij}$ (recall that $x_{ij} \in \{0,1\}$), the energy function can be rewritten as:

$$E'(x) = c'^{T}x + \frac{1}{2}x^{T}Q'x \qquad \text{Equation 8.7}$$

where $c' = c + \left(\frac{\mu}{2}\right)e$ and $Q' = Q - \mu I$ is a symmetric negative definite matrix. (See Appendix F for a proof that Q' is negative definite.) Note that $E(x) = E'(x)$ for all $x_{ij} \in \{0,1\}$.

The global minimum of a strictly concave quadratic function is attained at an extreme point of the feasible polytope. By relaxing the integer restrictions in the polytope P above to get the new polytope P′ consisting of the constraints (Equation 8.1) and (Equation 8.2), and the bounds $0 \le x_{ij} \le 1$ for i=1,...,n and j=1,...,N, it can be seen that the extreme points of P′ correspond to the feasible points of P. Hence, a global minimum for the strictly concave quadratic problem:

$$\min_{x \in P'} E'(x) \qquad \text{Equation 8.8}$$

will also be a global minimum of the discrete quadratic assignment problem (Equation 8.6).

Thus, the original quadratic assignment problem is equivalent to the minimization of a concave quadratic function on the unit hypercube with nN variables, each of which is restricted to the interval [0,1]. Two different computational methods for this class of problem have recently been developed. Either of these methods, linear underestimating functions or a stochastic method solving multiple cost row linear programs, can be used to provide the minimum of the concave, quadratic function. The global or near global solution to this problem provides a convenient starting point for the relaxed, local minimization problem.

The solution of the quadratic assignment problem provides the conformation with the lowest potential energy over all possible conformations *on the lattice*. Since this conformation may not lie at the exact position of the true global minimum, a local minimization procedure can be used to find the true solution. This minimization is therefore an unconstrained local minimization with 3(n-1) variables (Recall that $\theta_1=\varphi_1=\varphi_2=0$ are fixed.) over a continuous potential energy function (See Chapter II.). Any of the local minimization methods described in Chapter VI, including the quasi-Newton method, will satisfy the requirements of this problem.

D. Potential for Lattice Based Solutions

Unfortunately, any minimization based on the quadratic assignment method is totally reliant on the selection of an appropriate lattice. The establishment of a lattice which can accurately model the system quickly forces the problem to become intractable as problem size increases. Even for a very small problem, the construction of a lattice which can accurately represent the molecule in three-dimensional space incurs a very large number of variables (See Table 8.1.). Computing a solution based on such a large

Table 8.1 Number of Variables Required for Quadratic Assignment Formulation

	Lattice Size			
Number of Beads	5x5x5	6x6x6	7x7x7	10x10x10
5	625	1080	1715	5000
6	750	1296	2058	6000
7	875	1512	2401	7000
12	1500	2592	4116	12000

number of variables is currently intractable. Even the simplest problem described in Table 8.1 (5 beads on a 5x5x5 lattice), required 131 minutes to produce a minimum energy configuration on the lattice. Once this solution is given, the local minimization phase must still be run, and the final configuration remains *heavily* dependent on the selection of the lattice. As problem size increases, the number of variables required to form a suitable lattice increases at a rate which can be described by $O(n^4)$. The 625 variable problem incurred by the 5 bead hydrocarbon already approaches the limits of current optimization methods. Without prior knowledge of the protein structure (upon which an appropriate lattice could be based), the quadratic assignment formulation will not (Phillips and Rosen 1994) provide satisfactory results.

IX. Simulated Annealing

A. Optimization Analogous to Physical Annealing

Optimization by simulated annealing is one of the newest and most promising methods of finding the minimum or maximum of a function with multiple local optima. Simulated annealing is analogous to the physical process of annealing in condensed matter physics. In physical annealing, a solid is heated until it melts. It is then carefully cooled until it crystallizes into a state with a perfect lattice. During this process the free energy of the solid is minimized. Likewise, simulated annealing attempts to optimize a function by reducing the "temperature" of the solution's potential function until a global minimum energy is reached (Laarhoven and Aarts 1989).

B. Simulated Annealing Theory

The name "simulated annealing" originated from the analogy with the physical annealing process of solids. Annealing consists of two steps (Aarts and Korst 1989):

- Increase the temperature to a maximum value at which time the solid melts.
- Decrease **carefully** the temperature until the particles arrange themselves into the ground state of the solid.

The solid shifts from a state in which all particles arrange themselves randomly, to a ground state in which the particles are arranged in a highly structured lattice. In this lattice state, the energy of the system is minimal. Thus, the ground state is obtained only if the maximum temperature is sufficiently high, and the cooling is done sufficiently slow.

A computational equivalent of simulated annealing can be based on the following ideas:

- Solutions in a combinatorial optimization problem are equivalent to

states in a physical system.

- The cost of a solution is equivalent to the energy of a state.

The role of temperature is represented by a control parameter, c. If i and j represent two solutions to a combinatorial optimization problem, with corresponding costs f(i), and f(j), then the probability that solution j is accepted after visiting solution i is:

$$\begin{cases} 1 & \text{if } f(j) \le f(i) \\ e^{\left(\frac{f(i)-f(j)}{c}\right)} & \text{if } f(j) > f(i) \end{cases}.$$

For every solution generated, this acceptance criterion is applied, giving simulated annealing the unique feature of being able to accept limited deteriorations in potential value. The simulated annealing algorithm can be stated as:

1. Given some initial configuration, ϕ_0, with corresponding control parameter, c_0, and number of transitions, L_k and set $k \leftarrow 0$ and $f(\phi) = F(\phi_0)$.
2. Set $t \leftarrow 1$.
3. Generate a new point ϕ_k, with potential, f(ϕ_k).
4. If $f(\phi_k) < f(\phi)$ then $\phi = \phi_k$.
5. Otherwise, if $e^{\left(\frac{f(i)-f(j)}{c_k}\right)}$ is greater than a random number on the interval [0,1) then $\phi = \phi_k$.
6. Set $t \leftarrow t + 1$ and if t is less than L_k go back to step 2.
7. Calculate new number of transitions, L_k.
8. Calculate new control parameter, c_k.
9. Return to step 2.

In this algorithm, c_k is the value of the control parameter, and L_k the number of transitions

generated at the k^{th} iteration of the algorithm (Aarts and Korst 1989). The generation of the next set of parameters, represented by step three above is done by applying a perturbation mechanism. This mechanism transforms the current state into the next state by a small distortion. In the analogy with the physical process of annealing, this change may be the displacement of a particle. In combinatorial optimization, this change represents slight parameter alterations.

Simulated annealing is essentially a generalization of a local search. When the control parameter is set to zero, simulated annealing is identical to a local search algorithm, since the value of the potential function is only allowed to decrease. However, the ability to move through ranges of higher potential function values enables simulated annealing to find high quality solutions which do not strongly depend on the initial solution. Thus, simulated annealing has the capability to:

- process potential functions possessing arbitrary degrees of nonlinearities, discontinuities, and stochasticity.
- process arbitrary boundary conditions and constraints imposed on these potential functions.
- be implemented with a minimal degree of coding relative to other nonlinear optimization algorithms.
- statistically guarantee the finding of an optimal solution (ergodicity). (Ingber 1993)

C. Simulated Annealing and the Protein Folding Problem

Implementations of simulated annealing on the protein folding problem did not meet expected results. An implementation of the simulated annealing algorithm on the trivially simple 4 bead hydrocarbon, butane (containing three local minima) failed to return the global minimum after several thousand function evaluations. Simulated annealing requires the accurate selection of temperature schedules, control parameters, and stop-

ping criteria. With no knowledge of the potential function structure or complexity, selection of these values becomes extremely difficult. To further complicate the situation, the user must provide the perturbation mechanism used to generate new points. Even extremely simple problems, such as a concave quadratic, become very slow if simulated annealing is not set up correctly. For instance, even if the global minimum was selected on the first attempt, the simulated annealing algorithm will still spend significant amounts of time "bouncing" around the minimum point, occasionally allowing increases in function value, only to insure that the true global minimum has been found. With correct parameter choices, the simulated annealing algorithm may be able to make this decision quickly, but without prior knowledge of the function, simulated annealing's sensitivity to parameter changes makes its use prohibitive.

X. Conclusions

Three computational methods for solving the protein folding problem were examined. Two of these methods failed to produce acceptable results. A lattice based method, which allows the formulation of a quadratic assignment problem, failed due to the immense number of variables required to accurately model a protein chain of more than trivial length (more than five amino acids). Solving such a problem, containing a very large number of variables, is currently computationally intractable.

Simulated annealing has the theoretical ability to generate solutions to the problem, but is completely dependent on the selection of parameters which affect its operation. Without knowledge of the characteristics of the potential energy function, the selection of these values is very difficult.

The only successful method tested was the convex quadratic global underestimator. Using this technique, predictions for n-chain hydrocarbons of length 5 through 15 and 22 were made. However, large increases in local minimization times make predictions for chains of length greater than 22 unlikely using the current technique. The molecules tested produced a least squares fit of $0.23n^3 - 4.88n^2 + 36.03n - 85.8836$ for total prediction time (minutes). Using this approximating function, computation of an protein sized molecule (100 residues), would take 133 days of CPU time. For the underestimator to realize its potential, a superior method of finding local minima must be developed. The local minimization procedures rely on many function evaluations. Improvements in the function evaluation time (including conversions) would be of immense benefit, and perhaps provide the key to reducing local minimization times. Recent work by Greengard (1988) has attempted to specifically address this problem.

XI. References

Aarts, E., and Korst, J. (1989), Simulated Annealing and Boltzman Machines, John Wiley and Sons, New York.

Abagyan, R.A. (1993), Towards Protein Folding by Global Energy Optimization, Federation of European Biochemical Societies: Letters 325(1,2), 17-22.

Baker, D., Sohl, J.L., and Agard, D.A. (1992) Nature 356, 263-265.

Benner, S.A. and Gerloff, D.L. (1993), Predicting the Conformation of Proteins, Federation of European Biochemical Societies: Letters 325(1,2), 29-33.

Case, D.A. (1993), Computer Simulations of Protein Dynamics and Thermodynamics, IEEE: Computer Science & Engineering, October, 47-57.

Fletcher, R. (1987), Practical Methods of Optimization, John Wiley and Sons, New York.

Ferguson, D.M., Marsh, A., Metzger, T., Garret, D., and Kastella, K. (1994), Conformational Searches for the Global Minimum of Protein Models, Journal of Global Optimization 4, 209-227.

Gill, P.E., Murray, W., and Wright, M.H. (1981), Practical Optimization, Academic Press Inc., London.

Greengard, L. (1988), The Rapid Evaluation of Potential Fields in Particle Systems, The MIT Press, Cambridge.

Hinds, D.A. and Levitt, M. (1992), A Lattice Model for Protein Structure Prediction at Low Resolution, Proceedings of the National Academy of Science USA 89, 2536-2540.

Ingber, L. (1993), Simulated Annealing Practice Versus Theory, Manuscript.

Laarhoven, P.J.M. and Aarts, E. (1989), Simulated Annealing Theory and Applications, Kluwer Academic Publishers, Boston.

Lehninger, A.L. (1970), Biochemistry: The Molecular Basis of Cell Structure and Function, Worth Publishers, New York.

Maranas, C.D. and Floudas, C.A. (1994), Global Minimum Potential Energy Conformations of Small Molecules, Journal of Global Optimization 4(2), 135-170.

Phillips, A.T. and Rosen, J.B. (1994), A Quadratic Assignment Formulation of the Protein Folding Problem, Journal of Global Optimization 4(2), 229-241.

Richards, F.M. (1991), The Protein Folding Problem, Scientific American, 54-63.

Surles, M.C. (1992), An Algorithm With Linear Complexity For Interactive, Physically-based Modeling of Large Proteins, Computer Graphics 26, 221-230.

Swope, W.C. and Ferguson, D.M. (1992), Alternative Expressions for Energies and Forces Due to Angle Bending and Torsional Energy, Journal of Computational Chemistry 13(5), 585-594.

XII. Appendices

A. Conformers for Heptane

Table 12.1 Local Minima from Preferred Angles, Heptane

U(ϕ)	Initial Position, $\phi^{(0)}$				Minimized Position, ϕ^{*}			
-0.339727	180	180	180	180	180.000000	180.000000	180.000000	180.000000
-0.296889	180	-60	180	180	179.692157	-62.430409	179.609519	180.116728
-0.296802	180	180	-60	180	180.123730	179.592108	-62.493410	179.681464
-0.296761	180	180	60	180	179.863753	180.474235	62.478310	180.373030
-0.296694	180	60	180	180	180.309366	62.458289	180.392369	179.882561
-0.235524	180	60	60	180	180.794776	63.870585	63.876044	180.794720
-0.235466	180	-60	-60	180	179.194198	-63.899963	-63.877309	179.194518
-0.215609	-60	60	-60	60	-62.657878	179.600918	-179.857154	179.956953
-0.215524	-60	180	180	180	-62.706316	179.627701	180.119720	180.000000
-0.215508	180	60	-60	60	180.037859	179.841422	-179.602557	62.704924
-0.215492	60	180	180	180	62.682993	180.367282	179.881415	180.000000
-0.215452	180	180	180	-60	180.000000	180.121010	179.630935	-62.689425
-0.215452	180	180	180	60	180.000000	179.850405	180.407944	62.646775
-0.195729	180	-60	180	-60	179.718991	-62.709134	178.586693	-62.896149
-0.195689	-60	180	-60	180	-62.895246	178.613080	-62.715267	179.718298
-0.195639	60	180	60	180	62.919639	181.405828	62.707925	180.299240
-0.195567	60	-60	60	-60	62.908155	-178.598808	62.670638	-179.715697
-0.195526	180	60	180	60	180.321668	62.645896	181.385888	62.895050
-0.139428	60	180	180	-60	62.500992	180.750770	179.249089	-62.499945
-0.139423	-60	180	180	60	-62.475940	179.259740	180.740408	62.475995
-0.125230	60	60	180	180	64.173448	63.997801	180.864636	179.735869

Table 12.1 Local Minima from Preferred Angles, Heptane

$U(\phi)$	Initial Position, $\phi^{(0)}$				Minimized Position, ϕ^*			
-0.125181	-60	60	-60	-60	-179.762381	179.114165	-63.985617	-64.185171
-0.125144	180	180	60	60	179.760412	180.856949	63.972239	64.164748
-0.125125	-60	-60	180	180	-64.177304	-64.007009	179.159146	180.248780
-0.125125	-60	-60	60	-60	-64.210703	-64.018941	179.154209	-179.752857
-0.125101	180	180	-60	-60	180.222763	179.150008	-64.035518	-64.156472
-0.124850	-60	180	60	180	-62.737240	180.041359	62.538518	180.313344
-0.124826	60	180	-60	180	62.766715	179.950797	-62.544082	179.694691
-0.124723	180	-60	180	60	179.676542	-62.513659	179.959773	62.721797
-0.124681	180	60	180	-60	180.321851	62.542211	180.033777	-62.749200
-0.115072	-60	-60	180	-60	-63.801675	-64.208098	177.402217	-62.874698
-0.115043	60	180	60	60	62.902069	182.598851	64.182034	63.817398
-0.115031	60	60	180	60	63.819117	64.207828	182.636566	62.861170
-0.114919	-60	180	-60	-60	-62.897443	177.358042	-64.224387	-63.846126
-0.104232	60	180	180	60	62.576398	180.185008	180.185012	62.575741
-0.104125	-60	180	180	-60	-62.534582	179.808088	179.807766	-62.603440
-0.104118	60	60	-60	60	62.550498	180.191468	-179.798849	62.592457
0.009673	180	60	60	60	180.725871	63.726114	64.947038	64.047553
0.009674	60	60	60	180	64.016389	64.958687	63.664674	180.702491
0.009691	180	-60	-60	-60	179.273906	-63.686327	-64.964465	-64.041972
0.009736	-60	-60	-60	180	-64.055443	-64.987973	-63.684967	179.269252
0.045243	60	180	-60	-60	62.685626	179.523486	-64.182031	-64.199601
0.045296	-60	-60	180	60	-64.176786	-64.166840	179.514057	62.669550
0.045313	-60	180	60	60	-62.666795	180.526585	64.165301	64.223136
0.045388	60	60	180	-60	64.209791	64.193968	180.526278	-62.685254
0.249514	60	60	60	60	64.088236	64.918552	64.907186	64.083887
0.249592	-60	-60	-60	-60	-64.088099	-64.893060	-64.880987	-64.070387

Table 12.1 Local Minima from Preferred Angles, Heptane

$U(\phi)$	Initial Position, $\phi^{(0)}$				Minimized Position, ϕ^*			
2.026672	60	60	-60	-60	63.000643	79.493002	-79.562918	-62.995198
2.026839	-60	-60	60	60	-62.996259	-79.546417	79.567594	62.981162
2.058274	-60	60	60	180	-80.186256	78.462644	62.066901	181.075020
2.058303	60	-60	-60	180	80.184791	-78.395394	-62.095009	178.921228
2.058343	180	60	60	-60	181.106215	62.088491	78.404974	-80.180360
2.058378	180	-60	-60	60	178.917917	-62.052305	-78.445940	80.186494
2.070151	180	-60	60	180	179.287701	-79.399090	79.614853	180.690834
2.070219	180	60	-60	180	180.714220	79.494318	-79.365457	179.293132
2.097911	180	60	-60	-60	180.595288	79.299839	-79.307316	-62.430627
2.097949	-60	-60	60	180	-62.396732	-79.336814	79.286330	180.623456
2.097992	180	-60	60	60	179.378780	-79.278332	79.236470	62.423103
2.098034	60	60	-60	180	62.411331	79.279927	-79.366127	179.373374
2.158417	180	180	-60	60	180.225655	179.209800	-79.355190	79.491831
2.158423	-60	60	180	180	-79.398437	79.611536	180.835907	179.812488
2.158439	180	180	60	-60	179.816175	180.791002	79.434474	-79.543133
2.158467	60	-60	180	180	79.514152	-79.455677	179.243947	180.205755
2.230068	60	180	60	-60	62.935288	182.279514	82.013853	-77.339439
2.230082	-60	60	180	60	-77.265866	82.102048	182.245218	62.944156
2.230124	60	-60	180	-60	77.368066	-81.914342	177.749227	-62.976229
2.230226	-60	180	-60	60	-62.947196	177.778962	-81.899852	77.352625
2.336922	60	60	60	-60	64.085833	63.375309	78.562093	-79.969246
2.337016	-60	60	60	60	-79.963708	78.570336	63.378969	64.079620
2.337018	60	-60	-60	-60	79.875053	-78.515297	-63.341112	-64.066549
2.337127	-60	-60	-60	60	-64.098840	-63.359511	-78.518880	79.935753
2.343269	-60	60	180	-60	-79.299852	79.719299	180.326332	-62.698935
2.343326	60	180	-60	60	62.700984	179.692084	-79.670324	79.291646

Table 12.1 Local Minima from Preferred Angles, Heptane

$U(\phi)$	Initial Position, $\phi^{(0)}$				Minimized Position, ϕ^*			
2.343330	60	-60	180	60	79.431571	-79.717732	179.674046	62.754099
2.343345	-60	180	60	-60	-62.719402	180.352229	79.647337	-79.369962
4.151268	60	-60	-60	60	78.132822	-76.957224	-76.943581	78.178186
4.151353	-60	60	60	-60	-78.142142	76.959901	76.950608	-78.166430
4.384072	180	-60	60	-60	178.741629	-93.723491	69.284953	-94.158124
4.384075	-60	60	-60	180	-94.198578	69.340226	-93.775408	178.766022
4.384250	60	-60	60	180	94.299939	-69.383118	93.736766	181.247700
4.502235	60	-60	60	60	95.723592	-70.556034	90.568777	61.256347

B. Global Underestimator Run: Butane (4 beads)

Host Machine: haney Tue Apr 12 10:46:53 EDT 1994

How many randomizations (at least 3): 10

Within what radius of phi[1] = 180.0000 (degs): 180

--> Using 10 randomizations

running QN-BFGS (using all local minima) ... pass #1

Elapsed Real Time for QN-BFGS = 1.000 Seconds

--> 11 points reduced to 11 unique conformers

solving LP ...

Conformations: 11 (possibly non-unique)
Dihedral Angles: 1
Quad Fit Error = 0.014714 kcal/mol (sum of errors)
LP Pivots = 4

Interpolated Conformations:

1 2 3 (3 out of 11)

Convex Quadratic Underestimator:

c[0] = +0.553564
c[1] = -0.357722 d[1] = +0.113864

Max error in Quad Fit = 0.006454 kcal/mol

Distribution of Errors (% of Max)

10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
8	0	0	1	0	0	0	0	1	1

Ranges on the Dihedrals (Degrees):

63.011900 <= phi[1] <= 297.081800

Elapsed Real Time for LP = 0.000 Seconds

LP Predicted Dihedrals Current Global Minimum

phi[1] = 3.141660 rads (180.003872 degs) phi[1] = 3.141593 rads (180.000000 degs)

Energy = -0.008356 kcal/mol

LP Prediction/Global Minimum Difference (inf_norm) = 0.000068 degs
LP Prediction/Global Minimum Quad Function Difference = 0.000000 kcal/mol

---
Repeat (y/n)? y

How many randomizations: 5
What fraction of the distance to the global min should be used [0,1]: 0
Use how many points from previous run (global only/all): g

Within what radius of phi[1] = 180.0039 (degs): 5

--> Using 5 randomizations

running QN-BFGS (using only global minimum) ... pass #2

Elapsed Real Time for QN-BFGS = 1.000 Seconds

--> 7 points reduced to 5 unique conformers

solving LP ...

Conformations: 5 (possibly non-unique)

Dihedral Angles: 1
Quad Fit Error = 0.000681 kcal/mol (sum of errors)
LP Pivots = 6

Interpolated Conformations:

3 4 (2 out of 5)

Convex Quadratic Underestimator:

c[0] = +0.221661
c[1] = -0.073250 d[1] = +0.000000

Max error in Quad Fit = 0.000348 kcal/mol

Distribution of Errors (% of Max)

10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
2	0	0	1	0	0	1	0	0	1

Ranges on the Dihedrals (Degrees):

179.196100 <= phi[1] <= 180.156200

Elapsed Real Time for LP = 0.000 Seconds

LP Predicted Dihedrals Current Global Minimum

phi[1] = 3.141593 rads* (180.000000 degs) phi[1] = 3.141593 rads (180.000000 degs)

Energy = -0.008356 kcal/mol

LP Prediction/Global Minimum Difference (inf_norm) = 0.000000 degs
LP Prediction/Global Minimum Quad Function Difference = 0.000000 kcal/mol

---

Successive LP predicted dihedrals change (inf-norm) = 0.003900 degs

Successive global conformers change (inf-norm) = 0.000000 degs

Energy improvement = 0.000000 kcal/mol

Repeat (y/n)? n
Run Complete....

C. Global Underestimator Run: Heptane (7 beads)

Host Machine: haney Tue Apr 12 11:18:57 EDT 1994

How many randomizations (at least 9): 25

Within what radius of phi[1] = 20.0000 (degs): 180
Within what radius of phi[2] = 330.0000 (degs): 180
Within what radius of phi[3] = 278.0000 (degs): 180
Within what radius of phi[4] = -60.0000 (degs): 180

--> Using 25 randomizations

running QN-BFGS (using all local minima) ... pass #1

Elapsed Real Time for QN-BFGS = 40.000 Seconds

--> 26 points reduced to 26 unique conformers

solving LP ...

Conformations: 26 (possibly non-unique)
Dihedral Angles: 4
Quad Fit Error = 24.465463 kcal/mol (sum of errors)
LP Pivots = 13

Interpolated Conformations:

1 8 9 10 12 13 17 21 26 (9 out of 26)

Convex Quadratic Underestimator:

c[0] = +1.313334
c[1] = -0.299179 d[1] = +0.087703
c[2] = -0.273989 d[2] = +0.083520
c[3] = -0.225441 d[3] = +0.079214
c[4] = -0.344606 d[4] = +0.100744

Max error in Quad Fit = 4.677598 kcal/mol

Distribution of Errors (% of Max)

10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
17	0	0	0	7	0	0	0	0	2

Ranges on the Dihedrals (Degrees):

62.664800 <= phi[1] <= 297.534500
62.362800 <= phi[2] <= 297.492700
62.576500 <= phi[3] <= 297.954800
62.516700 <= phi[4] <= 297.321200

Elapsed Real Time for LP = 1.000 Seconds

LP Predicted Dihedrals | Current Global Minimum

phi[1] = 3.411274 rads (195.451627 degs) | phi[1] = 3.143848 rads (180.129200 degs)
phi[2] = 3.280520 rads (187.959930 degs) | phi[2] = 3.133486 rads (179.535500 degs)
phi[3] = 2.845974 rads (163.062310 degs) | phi[3] = 5.192398 rads (297.502500 degs)
phi[4] = 3.420611 rads (195.986554 degs) | phi[4] = 3.135182 rads (179.632700 degs)

Energy = -0.296754 kcal/mol

LP Prediction/Global Minimum Difference (inf_norm) = 2.346424 degs
LP Prediction/Global Minimum Quad Function Difference = 0.226207 kcal/mol

---

Repeat (y/n)? y

How many randomizations: 20
What fraction of the distance to the global min should be used [0,1]: 0
Use how many points from previous run (global only/all): g

Within what radius of phi[1] = 195.4516 (degs): 20
Within what radius of phi[2] = 187.9599 (degs): 15
Within what radius of phi[3] = 163.0623 (degs): 140
Within what radius of phi[4] = 195.9866 (degs): 20

--> Using 20 randomizations

running QN-BFGS (using only global minimum) ... pass #2

Elapsed Real Time for QN-BFGS = 15.000 Seconds

--> 22 points reduced to 22 unique conformers

solving LP ...

Conformations: 22 (possibly non-unique)
Dihedral Angles: 4
Quad Fit Error = 0.011400 kcal/mol (sum of errors)
LP Pivots = 20

Interpolated Conformations:

1 2 7 8 11 20 21 (7 out of 22)

Convex Quadratic Underestimator:

c[0] = +46.835326

c[1] = -0.056420 d[1] = +0.000000
c[2] = -0.051352 d[2] = +0.000000
c[3] = -0.063887 d[3] = +0.020319
c[4] = -29.772245 d[4] = +9.482895

Max error in Quad Fit = 0.002471 kcal/mol

Distribution of Errors (% of Max)

10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
10	3	2	2	3	1	0	0	0	1

Ranges on the Dihedrals (Degrees):

179.383400 <= phi[1] <= 180.446000
179.018300 <= phi[2] <= 180.749100
62.158100 <= phi[3] <= 297.544400
179.528500 <= phi[4] <= 181.045700

Elapsed Real Time for LP = 1.000 Seconds

LP Predicted Dihedrals Current Global Minimum

phi[1] = 3.141902 rads* (180.017700 degs) phi[1] = 3.141902 rads (180.017700 degs)
phi[2] = 3.141477 rads* (179.993400 degs) phi[2] = 3.141477 rads (179.993400 degs)
phi[3] = 3.144200 rads (180.149391 degs) phi[3] = 3.142771 rads (180.067500 degs)
phi[4] = 3.139573 rads (179.884306 degs) phi[4] = 3.141183 rads (179.976500 degs)

Energy = -0.339768 kcal/mol

LP Prediction/Global Minimum Difference (inf_norm) = 0.001609 degs
LP Prediction/Global Minimum Quad Function Difference = 0.000012 kcal/mol

---

Successive LP predicted dihedrals change (inf-norm) = 17.087100 degs

Successive global conformers change (inf-norm) = 117.435000 degs

Energy improvement = 0.043014 kcal/mol

Repeat (y/n)? y

How many randomizations: 10
What fraction of the distance to the global min should be used [0,1]: 0
Use how many points from previous run (global only/all): g

Within what radius of phi[1] = 180.0177 (degs): 4
Within what radius of phi[2] = 179.9934 (degs): 4
Within what radius of phi[3] = 180.1494 (degs): 5

Within what radius of phi[4] = 179.8843 (degs): 5

--> Using 10 randomizations

running QN-BFGS (using only global minimum) ... pass #3

Elapsed Real Time for QN-BFGS = 3.000 Seconds

--> 12 points reduced to 12 unique conformers

solving LP ...

Conformations: 12 (possibly non-unique)
Dihedral Angles:4
Quad Fit Error =0.000950 kcal/mol (sum of errors)
LP Pivots = 10

Interpolated Conformations:

2 3 4 7 10 12 (6 out of 12)

Convex Quadratic Underestimator:

c[0] = +88.986822
c[1] = -0.046979 d[1] = +0.000000
c[2] = -56.991488 d[2] = +18.152711
c[3] = +0.006854 d[3] = +0.000000
c[4] = +0.083894 d[4] = +0.000000

Max error in Quad Fit = 0.000436 kcal/mol

Distribution of Errors (% of Max)

10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
7	0	1	2	0	1	0	0	0	1

Ranges on the Dihedrals (Degrees):

179.194700 <= phi[1] <= 180.700300
178.951300 <= phi[2] <= 180.893700
179.722400 <= phi[3] <= 180.467900
179.851200 <= phi[4] <= 180.888000

Elapsed Real Time for LP = 0.000 Seconds

LP Predicted Dihedrals Current Global Minimum

phi[1] = 3.141902 rads* (180.017700 degs)
phi[2] = 3.139558 rads (179.883420 degs)
phi[3] = 3.143699 rads* (180.120700 degs)
phi[4] = 3.139963 rads* (179.906600 degs)

phi[1] = 3.141902 rads (180.017700 degs)
phi[2] = 3.141477 rads (179.993400 degs)
phi[3] = 3.143699 rads (180.120700 degs)
phi[4] = 3.139963 rads (179.906600 degs)

Energy = -0.339793 kcal/mol

LP Prediction/Global Minimum Difference (inf_norm) = 0.001920 degs
LP Prediction/Global Minimum Quad Function Difference = 0.000033 kcal/mol

---

Successive LP predicted dihedrals change (inf-norm) = 0.110000 degs

Successive global conformers change (inf-norm) = 0.069900 degs

Energy improvement = 0.000025 kcal/mol

Repeat (y/n)? y

How many randomizations: 5
What fraction of the distance to the global min should be used [0,1]: 0
Use how many points from previous run (global only/all): g

Within what radius of phi[1] = 180.0177 (degs): 0
Within what radius of phi[2] = 179.8834 (degs): 5
Within what radius of phi[3] = 180.1207 (degs): 0
Within what radius of phi[4] = 179.9066 (degs): 0

--> Using 5 randomizations

running QN-BFGS (using only global minimum) ... pass #4

Elapsed Real Time for QN-BFGS = 1.000 Seconds

--> 7 points reduced to 7 unique conformers

solving LP ...

Conformations: 7 (possibly non-unique)
Dihedral Angles: 4
Quad Fit Error = 0.001902 kcal/mol (sum of errors)
LP Pivots = 12

Interpolated Conformations:

1 2 3 4 7 (5 out of 7)

Convex Quadratic Underestimator:

c[0] = +6.716253
c[1] = -0.830429 d[1] = +0.000000
c[2] = -0.061182 d[2] = +0.000000
c[3] = -0.464009 d[3] = +0.000000
c[4] = -0.890460 d[4] = +0.000000

Max error in Quad Fit = 0.001696 kcal/mol

Distribution of Errors (% of Max)

10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
5	1	0	0	0	0	0	0	0	1

Ranges on the Dihedrals (Degrees):

179.979400 <= phi[1] <= 180.017700
179.261800 <= phi[2] <= 180.713700
179.971500 <= phi[3] <= 180.120700
179.906600 <= phi[4] <= 180.009200

Elapsed Real Time for LP = 0.000 Seconds

LP Predicted Dihedrals Current Global Minimum

phi[1] = 3.141902 rads* (180.017700 degs) phi[1] = 3.141902 rads (180.017700 degs)
phi[2] = 3.141477 rads* (179.993400 degs) phi[2] = 3.141477 rads (179.993400 degs)
phi[3] = 3.143699 rads* (180.120700 degs) phi[3] = 3.143699 rads (180.120700 degs)
phi[4] = 3.139963 rads* (179.906600 degs) phi[4] = 3.139963 rads (179.906600 degs)

Energy = -0.339793 kcal/mol

LP Prediction/Global Minimum Difference (inf_norm) = 0.000000 degs
LP Prediction/Global Minimum Quad Function Difference = 0.000000 kcal/mol

---

Successive LP predicted dihedrals change (inf-norm) = 0.110000 degs

Successive global conformers change (inf-norm) = 0.000000 degs

Energy improvement = 0.000000 kcal/mol

Repeat (y/n)? n
Run Complete....

D. Global Underestimator Run: 12 beads

Host Machine: csserverc Fri Apr 15 08:43:35 EDT 1994

How many randomizations (at least 19): 60

Within what radius of phi[1] = 330.0000 (degs): 180
Within what radius of phi[2] = 278.0000 (degs): 180
Within what radius of phi[3] = 123.0000 (degs): 180
Within what radius of phi[4] = 31.0000 (degs): 180
Within what radius of phi[5] = 160.0000 (degs): 180
Within what radius of phi[6] = 97.0000 (degs): 180
Within what radius of phi[7] = 27.0000 (degs): 180
Within what radius of phi[8] = 232.0000 (degs): 180
Within what radius of phi[9] = 38.0000 (degs): 180

--> Using 60 randomizations

running QN-BFGS (using all local minima) ... pass #1

Elapsed Real Time for QN-BFGS = 801.000 Seconds

--> 61 points reduced to 61 unique conformers

solving LP ...

Conformations: 61 (possibly non-unique)
Dihedral Angles: 9
Quad Fit Error = 84.557709 kcal/mol (sum of errors)
LP Pivots = 50

Interpolated Conformations:

2 5 13 20 21 23 25 29 31 33 37 39 42 55 57 61 (16 out of 61)

Convex Quadratic Underestimator:

c[0] = +4.689026

c[1] = -0.211345	d[1] = +0.083985
c[2] = -1.757095	d[2] = +0.538594
c[3] = -1.230106	d[3] = +0.284693
c[4] = +0.149471	d[4] = +0.000000
c[5] = -0.109763	d[5] = +0.000000
c[6] = -0.016002	d[6] = +0.025305
c[7] = +0.012777	d[7] = +0.000000
c[8] = -0.931717	d[8] = +0.188731
c[9] = +0.133952	d[9] = +0.003914

Max error in Quad Fit = 6.538950 kcal/mol

Distribution of Errors (% of Max)

10%	20%	30%	40%	50%	60%	70%	80%	90%	100%

26 14 3 7 1 6 0 1 2 1

Ranges on the Dihedrals (Degrees):

60.391500 <= phi[1] <= 297.814200
61.145700 <= phi[2] <= 297.677800
60.504700 <= phi[3] <= 299.337000
60.057200 <= phi[4] <= 299.964500
61.774000 <= phi[5] <= 301.509700
60.360500 <= phi[6] <= 300.002300
61.171200 <= phi[7] <= 298.399200
59.886200 <= phi[8] <= 299.800300
61.052100 <= phi[9] <= 298.280100

Elapsed Real Time for LP = 1.000 Seconds

LP Predicted Dihedrals	Current Global Minimum
phi[1] = 2.516461 rads (144.182610 degs)	phi[1] = 5.164642 rads (295.912200 degs)
phi[2] = 3.262374 rads (186.920255 degs)	phi[2] = 3.002010 rads (172.002500 degs)
phi[3] = 4.320816 rads (247.564507 degs)	phi[3] = 5.187037 rads (297.195300 degs)
phi[4] = 5.148753 rads* (295.001800 degs)	phi[4] = 5.148753 rads (295.001800 degs)
phi[5] = 2.881113 rads* (165.075600 degs)	phi[5] = 2.881113 rads (165.075600 degs)
phi[6] = 0.632365 rads (36.231854 degs)	phi[6] = 5.153449 rads (295.270900 degs)
phi[7] = 3.109494 rads* (178.160900 degs)	phi[7] = 3.109494 rads (178.160900 degs)
phi[8] = 4.936746 rads (282.854707 degs)	phi[8] = 3.139512 rads (179.880800 degs)
phi[9] = 3.475300 rads (199.120016 degs)	phi[9] = 1.065560 rads (61.052100 degs)
	Energy = -1.784457 kcal/mol

LP Prediction/Global Minimum Difference (inf_norm) = 4.521084 degs
LP Prediction/Global Minimum Quad Function Difference = 0.638772 kcal/mol

---

Repeat (y/n)? y

How many randomizations: 60
What fraction of the distance to the global min should be used [0,1]: 0
Use how many points from previous run (global only/all): g

Within what radius of phi[1] = 144.1826 (degs): 150
Within what radius of phi[2] = 186.9203 (degs): 20
Within what radius of phi[3] = 247.5645 (degs): 60
Within what radius of phi[4] = 295.0018 (degs): 180
Within what radius of phi[5] = 165.0756 (degs): 180
Within what radius of phi[6] = 36.2319 (degs): 160
Within what radius of phi[7] = 178.1609 (degs): 180
Within what radius of phi[8] = 282.8547 (degs): 120
Within what radius of phi[9] = 199.1200 (degs): 150

--> Using 60 randomizations

running QN-BFGS (using only global minimum) ... pass #2

Elapsed Real Time for QN-BFGS = 765.000 Seconds

--> 62 points reduced to 62 unique conformers

solving LP ...

Conformations: 62 (possibly non-unique)
Dihedral Angles: 9
Quad Fit Error = 81.330506 kcal/mol (sum of errors)
LP Pivots = 49

Interpolated Conformations:

1 3 7 15 17 18 19 20 22 26 33 35 36 43 49 60 (16 out of 62)

Convex Quadratic Underestimator:

c[0] = +1225.378595

c[1] = +0.120373	d[1] = +0.000000
c[2] = -809.515465	d[2] = +267.458814
c[3] = +0.397725	d[3] = +0.000000
c[4] = -0.388046	d[4] = +0.104867
c[5] = -0.103557	d[5] = +0.000000
c[6] = -0.816778	d[6] = +0.223116
c[7] = -0.821738	d[7] = +0.232574
c[8] = -0.183396	d[8] = +0.089597
c[9] = -1.261490	d[9] = +0.369553

Max error in Quad Fit = 4.766802 kcal/mol

Distribution of Errors (% of Max)

10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
22	11	4	0	14	5	1	0	2	3

Ranges on the Dihedrals (Degrees):

60.448900 <= phi[1] <= 298.329800
172.002500 <= phi[2] <= 182.659000
173.720700 <= phi[3] <= 300.268000
60.647700 <= phi[4] <= 300.452200
62.011400 <= phi[5] <= 298.101000
59.459000 <= phi[6] <= 300.846400
60.256700 <= phi[7] <= 299.676000
62.973100 <= phi[8] <= 301.085700
60.689600 <= phi[9] <= 298.120500

Elapsed Real Time for LP = 1.000 Seconds

LP Predicted Dihedrals	Current Global Minimum
phi[1] = 1.088288 rads* (62.354300 degs)	phi[1] = 1.088288 rads (62.354300 degs)
phi[2] = 3.026692 rads (173.416680 degs)	phi[2] = 3.093175 rads (177.225900 degs)
phi[3] = 5.181839 rads* (296.897500 degs)	phi[3] = 5.181839 rads (296.897500 degs)
phi[4] = 3.700363 rads (212.015201 degs)	phi[4] = 5.169088 rads (296.166900 degs)
phi[5] = 2.962707 rads* (169.750600 degs)	phi[5] = 2.962707 rads (169.750600 degs)
phi[6] = 3.660777 rads (209.747092 degs)	phi[6] = 5.171863 rads (296.325900 degs)
phi[7] = 3.533232 rads (202.439306 degs)	phi[7] = 5.200229 rads (297.951200 degs)
phi[8] = 2.046899 rads (117.278668 degs)	phi[8] = 3.078949 rads (176.410800 degs)
phi[9] = 3.413556 rads (195.582374 degs)	phi[9] = 3.161196 rads (181.123200 degs)
	Energy = -2.476850 kcal/mol

LP Prediction/Global Minimum Difference (inf_norm) = 1.666997 degs
LP Prediction/Global Minimum Quad Function Difference = 1.341558 kcal/mol

---

Successive LP predicted dihedrals change (inf-norm) = 173.515200 degs

Successive global conformers change (inf-norm) = 233.557900 degs

Energy improvement = 0.692393 kcal/mol

Repeat (y/n)? y

How many randomizations: 50
What fraction of the distance to the global min should be used [0,1]: .5
Use how many points from previous run (global only/all): g

Within what radius of phi[1] = 62.3543 (degs): 90
Within what radius of phi[2] = 175.3213 (degs): 10
Within what radius of phi[3] = 296.8975 (degs): 20
Within what radius of phi[4] = 254.0911 (degs): 50
Within what radius of phi[5] = 169.7506 (degs): 180
Within what radius of phi[6] = 253.0365 (degs): 50
Within what radius of phi[7] = 250.1953 (degs): 50
Within what radius of phi[8] = 146.8447 (degs): 50
Within what radius of phi[9] = 188.3528 (degs): 20

--> Using 50 randomizations

running QN-BFGS (using only global minimum) ... pass #3

Elapsed Real Time for QN-BFGS = 526.000 Seconds

--> 52 points reduced to 52 unique conformers

solving LP ...

Conformations: 52 (possibly non-unique)
Dihedral Angles: 9
Quad Fit Error = 57.460679 kcal/mol (sum of errors)
LP Pivots = 37

Interpolated Conformations:

1 5 19 20 24 26 29 31 35 36 38 40 48 (13 out of 52)

Convex Quadratic Underestimator:

c[0] = +15577.752075

c[1] = -1.572693	d[1] = +0.545076
c[2] = -25.942274	d[2] = +0.000000
c[3] = -5963.287255	d[3] = +1147.053749
c[4] = +0.145274	d[4] = +0.000000
c[5] = -5.584252	d[5] = +1.658767
c[6] = +0.321134	d[6] = +0.000000
c[7] = +0.015908	d[7] = +0.000000
c[8] = -1.244513	d[8] = +0.000000
c[9] = +4.039473	d[9] = +0.000000

Max error in Quad Fit = 3.752148 kcal/mol

Distribution of Errors (% of Max)

10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
21	5	4	3	1	10	3	1	1	3

Ranges on the Dihedrals (Degrees):

61.855600 <= phi[1] <= 299.884300
174.944100 <= phi[2] <= 180.196500
295.252600 <= phi[3] <= 301.388600
173.238200 <= phi[4] <= 298.521300
62.177300 <= phi[5] <= 297.667100
173.984000 <= phi[6] <= 298.358300
173.549100 <= phi[7] <= 298.672900
62.715800 <= phi[8] <= 182.877300
179.081600 <= phi[9] <= 181.467600

Elapsed Real Time for LP = 1.000 Seconds

LP Predicted Dihedrals Current Global Minimum

phi[1] = 2.885273 rads (165.313959 degs)
phi[2] = 3.100785 rads* (177.661900 deg)
phi[3] = 5.198786 rads (297.868511 degs)
phi[4] = 5.161232 rads* (295.716800 degs)
phi[5] = 3.366508 rads (192.886687 degs)
phi[6] = 5.160511 rads* (295.675500 degs)
phi[7] = 5.189426 rads* (297.332200 degs)
phi[8] = 3.092086 rads* (177.163500 degs)
phi[9] = 3.144113 rads* (180.144400 degs)

phi[1] = 3.146818 rads (180.299400 degs)
phi[2] = 3.100785 rads (177.661900 degs)
phi[3] = 5.191986 rads (297.478900 degs)
phi[4] = 5.161232 rads (295.716800 degs)
phi[5] = 2.960618 rads (169.630900 degs)
phi[6] = 5.160511 rads (295.675500 degs)
phi[7] = 5.189426 rads (297.332200 degs)
phi[8] = 3.092086 rads (177.163500 degs)
phi[9] = 3.144113 rads (180.144400 degs)

Energy = -3.006272 kcal/mol

LP Prediction/Global Minimum Difference (inf_norm) = 0.405890 degs
LP Prediction/Global Minimum Quad Function Difference = 0.181801 kcal/mol

---

Successive LP predicted dihedrals change (inf-norm) = 102.959700 degs

Successive global conformers change (inf-norm) = 117.945100 degs

Energy improvement = 0.529422 kcal/mol

Repeat (y/n)? y

How many randomizations: 40
What fraction of the distance to the global min should be used [0,1]: 0
Use how many points from previous run (global only/all): g

Within what radius of phi[1] = 165.3140 (degs): 20
Within what radius of phi[2] = 177.6619 (degs): 5
Within what radius of phi[3] = 297.8685 (degs): 5
Within what radius of phi[4] = 295.7168 (degs): 30
Within what radius of phi[5] = 192.8867 (degs): 40
Within what radius of phi[6] = 295.6755 (degs): 30
Within what radius of phi[7] = 297.3322 (degs): 30
Within what radius of phi[8] = 177.1635 (degs): 30
Within what radius of phi[9] = 180.1444 (degs): 5

--> Using 40 randomizations

running QN-BFGS (using only global minimum) ... pass #4

Elapsed Real Time for QN-BFGS = 404.000 Seconds

--> 42 points reduced to 42 unique conformers

solving LP ...

Conformations: 42 (possibly non-unique)
Dihedral Angles: 9
Quad Fit Error = 0.014822 kcal/mol (sum of errors)
LP Pivots = 59

Interpolated Conformations:

4 7 12 13 19 25 30 31 35 38 40 42 (12 out of 42)

Convex Quadratic Underestimator:

c[0] = +325.080069	
c[1] = +0.008313	d[1] = +0.000000
c[2] = -135.613593	d[2] = +43.781435
c[3] = +0.110036	d[3] = +0.000000
c[4] = +0.065432	d[4] = +0.000000
c[5] = +0.011291	d[5] = +0.000000
c[6] = +0.040999	d[6] = +0.000000
c[7] = +0.107349	d[7] = +0.000000
c[8] = -77.185222	d[8] = +24.898707
c[9] = -0.049789	d[9] = +0.000000

Max error in Quad Fit = 0.002218 kcal/mol

Distribution of Errors (% of Max)

10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
25	8	1	2	2	0	1	1	0	2

Ranges on the Dihedrals (Degrees):

179.779000 <= phi[1] <= 180.801700
177.013000 <= phi[2] <= 178.108200
297.070900 <= phi[3] <= 297.976400
295.240900 <= phi[4] <= 296.027200
169.495500 <= phi[5] <= 170.144400
295.366900 <= phi[6] <= 296.239900
297.076300 <= phi[7] <= 297.772100
176.903600 <= phi[8] <= 178.036600
179.817100 <= phi[9] <= 180.388500

Elapsed Real Time for LP = 1.000 Seconds

LP Predicted Dihedrals	Current Global Minimum
phi[1] = 3.144677 rads* (180.176700 degs)	phi[1] = 3.144677 rads (180.176700 degs)
phi[2] = 3.097514 rads (177.474460 degs)	phi[2] = 3.096150 rads (177.396300 degs)
phi[3] = 5.191054 rads* (297.425500 degs)	phi[3] = 5.191054 rads (297.425500 degs)
phi[4] = 5.162241 rads* (295.774600 degs)	phi[4] = 5.162241 rads (295.774600 degs)
phi[5] = 2.965339 rads* (169.901400 degs)	phi[5] = 2.965339 rads (169.901400 degs)

phi[6] = 5.162111 rads* (295.767200 degs)
phi[7] = 5.190196 rads* (297.376300 degs)
phi[8] = 3.099969 rads (177.615145 degs)
phi[9] = 3.144771 rads* (180.182100 degs)

phi[6] = 5.162111 rads (295.767200 degs)
phi[7] = 5.190196 rads (297.376300 degs)
phi[8] = 3.096898 rads (177.439200 degs)
phi[9] = 3.144771 rads (180.182100 degs)

Energy = -3.007255 kcal/mol

LP Prediction/Global Minimum Difference (inf_norm) = 0.003071 degs
LP Prediction/Global Minimum Quad Function Difference = 0.000158 kcal/mol

---

Successive LP predicted dihedrals change (inf-norm) = 22.985300 degs

Successive global conformers change (inf-norm) = 0.275700 degs

Energy improvement = 0.000983 kcal/mol

Repeat (y/n)? y

How many randomizations: 20
What fraction of the distance to the global min should be used [0,1]: 1
Use how many points from previous run (global only/all): g

Within what radius of phi[1] = 180.1767 (degs): 10
Within what radius of phi[2] = 177.3963 (degs): 10
Within what radius of phi[3] = 297.4255 (degs): 10
Within what radius of phi[4] = 295.7746 (degs): 10
Within what radius of phi[5] = 169.9014 (degs): 10
Within what radius of phi[6] = 295.7672 (degs): 10
Within what radius of phi[7] = 297.3763 (degs): 10
Within what radius of phi[8] = 177.4392 (degs): 10
Within what radius of phi[9] = 180.1821 (degs): 10

--> Using 20 randomizations

running QN-BFGS (using only global minimum) ... pass #5

Elapsed Real Time for QN-BFGS = 159.000 Seconds

--> 22 points reduced to 22 unique conformers

solving LP ...

Conformations: 22 (possibly non-unique)
Dihedral Angles: 9
Quad Fit Error = 0.008593 kcal/mol (sum of errors)
LP Pivots = 60

Interpolated Conformations:

1 3 4 7 8 9 12 13 19 20 (10 out of 22)

Convex Quadratic Underestimator:

c[0] = +0.000000

c[1] = +0.018242	d[1] = +0.000000
c[2] = +0.183987	d[2] = +0.000000
c[3] = +0.178514	d[3] = +0.000000
c[4] = +0.083497	d[4] = +0.000000
c[5] = -0.124721	d[5] = +0.000000
c[6] = -2.012263	d[6] = +0.407210
c[7] = +0.131898	d[7] = +0.000000
c[8] = -0.060945	d[8] = +0.000000
c[9] = -0.049604	d[9] = +0.000000

Max error in Quad Fit = 0.002296 kcal/mol

Distribution of Errors (% of Max)

10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
13	4	0	1	2	0	0		1	1

Ranges on the Dihedrals (Degrees):

179.959300 <= phi[1] <= 180.312200
177.251800 <= phi[2] <= 177.660500
297.214300 <= phi[3] <= 297.691900
295.571800 <= phi[4] <= 295.963200
169.559600 <= phi[5] <= 170.075500
295.394000 <= phi[6] <= 296.500900
297.208700 <= phi[7] <= 297.696600
177.086200 <= phi[8] <= 177.760500
180.000700 <= phi[9] <= 180.531600

Elapsed Real Time for LP = 0.000 Seconds

LP Predicted Dihedrals	Current Global Minimum
phi[1] = 3.144677 rads* (180.176700 degs)	phi[1] = 3.144677 rads (180.176700 degs)
phi[2] = 3.096150 rads* (177.396300 degs)	phi[2] = 3.096150 rads (177.396300 degs)
phi[3] = 5.191054 rads* (297.425500 degs)	phi[3] = 5.191054 rads (297.425500 degs)
phi[4] = 5.162241 rads* (295.774600 degs)	phi[4] = 5.162241 rads (295.774600 degs)
phi[5] = 2.965339 rads* (169.901400 degs)	phi[5] = 2.965339 rads (169.901400 degs)
phi[6] = 4.941585 rads (283.131989 degs)	phi[6] = 5.162111 rads (295.767200 degs)
phi[7] = 5.190196 rads* (297.376300 degs)	phi[7] = 5.190196 rads (297.376300 degs)
phi[8] = 3.096898 rads* (177.439200 degs)	phi[8] = 3.096898 rads (177.439200 degs)
phi[9] = 3.144771 rads* (180.182100 degs)	phi[9] = 3.144771 rads (180.182100 degs)
	Energy = -3.007255 kcal/mol

LP Prediction/Global Minimum Difference (inf_norm) = 0.220526 degs
LP Prediction/Global Minimum Quad Function Difference = 0.009902 kcal/mol

---

Successive LP predicted dihedrals change (inf-norm) = 12.635200 degs

Successive global conformers change (inf-norm) = 0.000000 degs

Energy improvement = 0.000000 kcal/mol

Repeat (y/n)? n
Run Complete....

E. Global Underestimator Run: 22 beads

Host Machine: csserverc Wed Apr 20 07:12:32 EDT 1994

How many randomizations (at least 39): 100

Within what radius of phi[1] = 330.0000 (degs): 180
Within what radius of phi[2] = 278.0000 (degs): 180
Within what radius of phi[3] = 123.0000 (degs): 180
Within what radius of phi[4] = 31.0000 (degs): 180
Within what radius of phi[5] = 160.0000 (degs): 180
Within what radius of phi[6] = 97.0000 (degs): 180
Within what radius of phi[7] = 27.0000 (degs): 180
Within what radius of phi[8] = 232.0000 (degs): 180
Within what radius of phi[9] = 38.0000 (degs): 180
Within what radius of phi[10] = 213.0000 (degs): 180
Within what radius of phi[11] = 311.0000 (degs): 180
Within what radius of phi[12] = 137.0000 (degs): 180
Within what radius of phi[13] = 28.0000 (degs): 180
Within what radius of phi[14] = 16.0000 (degs): 180
Within what radius of phi[15] = 323.0000 (degs): 180
Within what radius of phi[16] = 222.0000 (degs): 180
Within what radius of phi[17] = 289.0000 (degs): 180
Within what radius of phi[18] = 120.0000 (degs): 180
Within what radius of phi[19] = 179.0000 (degs): 180

--> Using 100 randomizations

running QN-BFGS (using all local minima) ... pass #1

Elapsed Real Time for QN-BFGS = 14465.000 Seconds

--> 99 points reduced to 99 unique conformers

solving LP ...

Conformations: 99 (possibly non-unique)
Dihedral Angles: 19
Quad Fit Error = 261.999901 kcal/mol (sum of errors)
LP Pivots = 126

Interpolated Conformations:

2 4 5 7 17 18 25 26 28 30 33 35 38 39 40 42 45 52 53 56 57 59 60 63 64 71 72 76 77 79 81 83 84 87 94 99 (36 out of 99)

Convex Quadratic Underestimator:

c[0] = +15.809699

c[1] = -0.238399	d[1] = +0.092840
c[2] = -1.555963	d[2] = +0.374113
c[3] = +0.041718	d[3] = +0.000000
c[4] = +0.320074	d[4] = +0.148104
c[5] = -0.523003	d[5] = +0.140036
c[6] = -1.399913	d[6] = +0.504272
c[7] = -2.284349	d[7] = +0.802268
c[8] = -0.884438	d[8] = +0.334472
c[9] = -0.529013	d[9] = +0.000000
c[10] = -1.135499	d[10] = +0.241783
c[11] = +0.393577	d[11] = +0.000000
c[12] = -1.486392	d[12] = +0.398304
c[13] = +0.095308	d[13] = +0.014153
c[14] = -0.221887	d[14] = +0.082623
c[15] = -2.938166	d[15] = +0.789750
c[16] = -2.479375	d[16] = +0.895932
c[17] = -1.420245	d[17] = +0.501218
c[18] = -0.300846	d[18] = +0.192963
c[19] = -1.454993	d[19] = +0.375462

Max error in Quad Fit = 10.712270 kcal/mol

Distribution of Errors (% of Max)

10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
43	10	8	12	9	4	6	3	2	2

Ranges on the Dihedrals (Degrees):

60.764300 <= phi[1] <= 300.277800
58.390200 <= phi[2] <= 301.986700
55.255200 <= phi[3] <= 300.553900
57.929100 <= phi[4] <= 302.100000
54.104100 <= phi[5] <= 302.815000
61.046500 <= phi[6] <= 304.852100
53.912900 <= phi[7] <= 309.291800
51.561400 <= phi[8] <= 305.891100
53.754300 <= phi[9] <= 307.740200
51.790700 <= phi[10] <= 307.421300
52.621200 <= phi[11] <= 310.690700
52.233600 <= phi[12] <= 305.259600

59.822800 <= phi[13] <= 306.844000
54.681400 <= phi[14] <= 303.379200
51.037900 <= phi[15] <= 303.680100
57.200900 <= phi[16] <= 304.314400
57.143800 <= phi[17] <= 301.433200
60.315100 <= phi[18] <= 300.888600
59.358800 <= phi[19] <= 300.052300

Elapsed Real Time for LP = 4.000 Seconds

LP Predicted Dihedrals	Current Global Minimum
phi[1] = 2.567848 rads (147.126848 degs)	phi[1] = 3.105243 rads (177.917300 degs)
phi[2] = 4.159072 rads (238.297287 degs)	phi[2] = 3.183406 rads (182.395700 degs)
phi[3] = 1.078123 rads* (61.771900 degs)	phi[3] = 1.078123 rads (61.771900 degs)
phi[4] = 4.122042 rads (236.175597 degs)	phi[4] = 3.164792 rads (181.329200 degs)
phi[5] = 3.734775 rads (213.986865 degs)	phi[5] = 3.149208 rads (180.436300 degs)
phi[6] = 2.776107 rads (159.059211 degs)	phi[6] = 1.088523 rads (62.367800 degs)
phi[7] = 2.847364 rads (163.141938 degs)	phi[7] = 3.071958 rads (176.010200 degs)
phi[8] = 2.644281 rads (151.506149 degs)	phi[8] = 3.145982 rads (180.251500 degs)
phi[9] = 5.148346 rads* (294.978500 degs)	phi[9] = 5.148346 rads (294.978500 degs)
phi[10] = 4.696356 rads (269.081368 degs)	phi[10] = 3.022744 rads (173.190500 degs)
phi[11] = 5.165801 rads* (295.978600 degs)	phi[11] = 5.165801 rads (295.978600 degs)
phi[12] = 3.731803 rads (213.816553 degs)	phi[12] = 5.196676 rads (297.747600 degs)
phi[13] = 5.832251 rads (334.163347 degs)	phi[13] = 5.129713 rads (293.910900 degs)
phi[14] = 2.685536 rads (153.869850 degs)	phi[14] = 1.678839 rads (96.190400 degs)
phi[15] = 3.720375 rads (213.161774 degs)	phi[15] = 1.144625 rads (65.582200 degs)
phi[16] = 2.767370 rads (158.558600 degs)	phi[16] = 3.288972 rads (188.444200 degs)
phi[17] = 2.833587 rads (162.352598 degs)	phi[17] = 1.174315 rads (67.283300 degs)
phi[18] = 1.559086 rads (89.329074 degs)	phi[18] = 3.072954 rads (176.067300 degs)
phi[19] = 3.875207 rads (222.033010 degs)	phi[19] = 3.208947 rads (183.859100 degs)
	Energy = -7.449145 kcal/mol

LP Prediction/Global Minimum Difference (inf_norm) = 2.575749 degs
LP Prediction/Global Minimum Quad Function Difference = 4.595416 kcal/mol

---

Repeat (y/n)? y

How many randomizations: 75
What fraction of the distance to the global min should be used [0,1]: 0
Use how many points from previous run (global only/all): g

Within what radius of phi[1] = 147.1268 (degs): 35
Within what radius of phi[2] = 238.2973 (degs): 55
Within what radius of phi[3] = 61.7719 (degs): 180
Within what radius of phi[4] = 236.1756 (degs): 55
Within what radius of phi[5] = 213.9869 (degs): 35
Within what radius of phi[6] = 159.0592 (degs): 100
Within what radius of phi[7] = 163.1419 (degs): 15

Within what radius of phi[8] = 151.5061 (degs): 30
Within what radius of phi[9] = 294.9785 (degs): 180
Within what radius of phi[10] = 269.0814 (degs): 100
Within what radius of phi[11] = 295.9786 (degs): 180
Within what radius of phi[12] = 213.8166 (degs): 85
Within what radius of phi[13] = 334.1633 (degs): 35
Within what radius of phi[14] = 153.8699 (degs): 60
Within what radius of phi[15] = 213.1618 (degs): 150
Within what radius of phi[16] = 158.5586 (degs): 30
Within what radius of phi[17] = 162.3526 (degs): 105
Within what radius of phi[18] = 89.3291 (degs): 90
Within what radius of phi[19] = 222.0330 (degs): 40

--> Using 75 randomizations

running QN-BFGS (using only global minimum) ... pass #2

Elapsed Real Time for QN-BFGS = 6746.000 Seconds

--> 77 points reduced to 77 unique conformers

solving LP ...

Conformations: 77 (possibly non-unique)
Dihedral Angles:19
Quad Fit Error =180.596223 kcal/mol (sum of errors)
LP Pivots = 159

Interpolated Conformations:

1 6 8 9 21 23 24 26 30 31 34 41 42 44 45 48 51 60 62 64 67 69 72 75 76 (25 out of 77)

Convex Quadratic Underestimator:

c[0] = +144.561300

c[1] = -34.584215	d[1] = +16.810994
c[2] = -13.561606	d[2] = +3.534313
c[3] = -0.939967	d[3] = +0.000000
c[4] = +0.845433	d[4] = +0.000000
c[5] = -31.655626	d[5] = +7.783377
c[6] = -0.457572	d[6] = +0.000000
c[7] = -5.701583	d[7] = +0.000000
c[8] = -1.826113	d[8] = +0.000000
c[9] = -0.438230	d[9] = +0.000000
c[10] = +0.490778	d[10] = +0.000000
c[11] = -2.679081	d[11] = +0.807040
c[12] = -0.166776	d[12] = +0.000000
c[13] = -0.366454	d[13] = +0.000000
c[14] = +0.147614	d[14] = +0.000000
c[15] = +0.104883	d[15] = +0.000000

c[16] = -3.023704 d[16] = +0.000000
c[17] = -0.722325 d[17] = +0.227437
c[18] = -0.905696 d[18] = +0.000000
c[19] = +0.756502 d[19] = +0.000000

Max error in Quad Fit = 9.905244 kcal/mol

Distribution of Errors (% of Max)

10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
30	6	10	12	7	7	3	1	0	1

Ranges on the Dihedrals (Degrees):

61.225300 <= phi[1] <= 181.832900
175.450700 <= phi[2] <= 301.786100
60.548900 <= phi[3] <= 301.842700
169.540400 <= phi[4] <= 299.865600
159.878900 <= phi[5] <= 299.666000
53.564400 <= phi[6] <= 303.470100
165.506600 <= phi[7] <= 207.251100
75.822700 <= phi[8] <= 205.201600
57.909800 <= phi[9] <= 307.113100
55.031500 <= phi[10] <= 311.931900
49.647200 <= phi[11] <= 301.752600
150.413300 <= phi[12] <= 311.839200
55.901100 <= phi[13] <= 308.383200
58.148900 <= phi[14] <= 198.021200
55.392300 <= phi[15] <= 305.325100
170.785700 <= phi[16] <= 193.669800
57.015500 <= phi[17] <= 302.169300
59.621500 <= phi[18] <= 183.323300
174.983100 <= phi[19] <= 297.404000

Elapsed Real Time for LP = 3.000 Seconds

LP Predicted Dihedrals	Current Global Minimum
phi[1] = 2.057238 rads (117.871053 degs)	phi[1] = 1.101978 rads (63.138700 degs)
phi[2] = 3.837126 rads (219.851153 degs)	phi[2] = 3.157618 rads (180.918200 degs)
phi[3] = 1.056778 rads* (60.548900 degs)	phi[3] = 1.056778 rads (60.548900 degs)
phi[4] = 3.172085 rads* (181.747100 degs)	phi[4] = 3.172085 rads (181.747100 degs)
phi[5] = 4.067081 rads (233.026586 degs)	phi[5] = 3.115551 rads (178.507900 degs)
phi[6] = 5.234387 rads* (299.908300 degs)	phi[6] = 5.234387 rads (299.908300 degs)
phi[7] = 3.154526 rads* (180.741000 degs)	phi[7] = 3.154526 rads (180.741000 degs)
phi[8] = 3.083152 rads* (176.651600 degs)	phi[8] = 3.083152 rads (176.651600 degs)
phi[9] = 5.186810 rads* (297.182300 degs)	phi[9] = 5.186810 rads (297.182300 degs)
phi[10] = 5.233930 rads* (299.882100 degs)	phi[10] = 5.233930 rads (299.882100 degs)
phi[11] = 3.319638 rads (190.201272 degs)	phi[11] = 3.052165 rads (174.876200 degs)
phi[12] = 5.244374 rads* (300.480500 degs)	phi[12] = 5.244374 rads (300.480500 degs)
phi[13] = 5.211585 rads* (298.601800 degs)	phi[13] = 5.211585 rads (298.601800 degs)

phi[14] = 3.067434 rads* (175.751000 degs)
phi[15] = 5.241103 rads* (300.293100 degs)
phi[16] = 3.108295 rads* (178.092200 degs)
phi[17] = 3.175934 rads (181.967639 degs)
phi[18] = 1.040591 rads* (59.621500 degs)
phi[19] = 3.149848 rads* (180.473000 degs)

phi[14] = 3.067434 rads (175.751000 degs)
phi[15] = 5.241103 rads (300.293100 degs)
phi[16] = 3.108295 rads (178.092200 degs)
phi[17] = 3.155606 rads (180.802900 degs)
phi[18] = 1.040591 rads (59.621500 degs)
phi[19] = 3.149848 rads (180.473000 degs)

Energy = -8.726705 kcal/mol

LP Prediction/Global Minimum Difference (inf_norm) = 0.955260 degs
LP Prediction/Global Minimum Quad Function Difference = 12.038637 kcal/mol

---

Successive LP predicted dihedrals change (inf-norm) = 140.849100 degs

Successive global conformers change (inf-norm) = 237.540500 degs

Energy improvement = 1.277560 kcal/mol

Repeat (y/n)? y

How many randomizations: 65
What fraction of the distance to the global min should be used [0,1]: 0.5
Use how many points from previous run (global only/all): g

Within what radius of phi[1] = 90.5049 (degs): 30
Within what radius of phi[2] = 200.3847 (degs): 20
Within what radius of phi[3] = 60.5489 (degs): 180
Within what radius of phi[4] = 181.7471 (degs): 60
Within what radius of phi[5] = 205.7672 (degs): 25
Within what radius of phi[6] = 299.9083 (degs): 60
Within what radius of phi[7] = 180.7410 (degs): 5
Within what radius of phi[8] = 176.6516 (degs): 10
Within what radius of phi[9] = 297.1823 (degs): 60
Within what radius of phi[10] = 299.8821 (degs): 10
Within what radius of phi[11] = 182.5387 (degs): 5
Within what radius of phi[12] = 300.4805 (degs): 60
Within what radius of phi[13] = 298.6018 (degs): 5
Within what radius of phi[14] = 175.7510 (degs): 60
Within what radius of phi[15] = 300.2931 (degs): 60
Within what radius of phi[16] = 178.0922 (degs): 10
Within what radius of phi[17] = 181.3853 (degs): 5
Within what radius of phi[18] = 59.6215 (degs): 60
Within what radius of phi[19] = 180.4730 (degs): 20

--> Using 65 randomizations

running QN-BFGS (using only global minimum) ... pass #3

Elapsed Real Time for QN-BFGS = 7484.000 Seconds

--> 67 points reduced to 67 unique conformers

solving LP ...

Conformations: 67 (possibly non-unique)
Dihedral Angles: 19
Quad Fit Error = 0.713424 kcal/mol (sum of errors)
LP Pivots = 124

Interpolated Conformations:

1 2 3 6 8 12 17 21 24 26 29 30 39 40 41 42 43 44 46 47 48 49 50 52 55 58 61 66 67 (29 out of 67)

Convex Quadratic Underestimator:

c[0] = +862.353679

c[1] = +1.576256	d[1] = +0.000000
c[2] = +11.505862	d[2] = +0.000000
c[3] = -0.755355	d[3] = +0.265304
c[4] = +0.861950	d[4] = +0.000000
c[5] = +1.905837	d[5] = +0.000000
c[6] = -14.157930	d[6] = +4.433968
c[7] = -8.344343	d[7] = +0.386241
c[8] = -3.335837	d[8] = +0.000000
c[9] = +3.106748	d[9] = +0.000000
c[10] = -68.281811	d[10] = +13.059085
c[11] = -8.726862	d[11] = +2.753363
c[12] = -3.075571	d[12] = +0.000000
c[13] = +3.623808	d[13] = +0.000000
c[14] = -22.205927	d[14] = +7.411516
c[15] = +0.243679	d[15] = +0.000000
c[16] = -31.923569	d[16] = +10.592633
c[17] = -0.710224	d[17] = +0.000000
c[18] = -5.924978	d[18] = +1.949442
c[19] = -391.635793	d[19] = +126.049962

Max error in Quad Fit = 0.073463 kcal/mol

Distribution of Errors (% of Max)

10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
41	6	6	6	5	0	1	0	1	1

Ranges on the Dihedrals (Degrees):

59.668300 <= phi[1] <= 180.233200
173.662000 <= phi[2] <= 186.797300
60.465200 <= phi[3] <= 302.153700
168.498000 <= phi[4] <= 302.173200
153.880200 <= phi[5] <= 297.450400
58.388000 <= phi[6] <= 306.578200

173.159200 <= phi[7] <= 302.506900
64.552300 <= phi[8] <= 294.232600
72.285500 <= phi[9] <= 303.774700
188.202400 <= phi[10] <= 305.621100
161.972500 <= phi[11] <= 294.907100
177.360700 <= phi[12] <= 306.147500
57.296400 <= phi[13] <= 301.846000
60.468000 <= phi[14] <= 293.192800
176.636700 <= phi[15] <= 304.215300
64.959300 <= phi[16] <= 301.386800
173.914200 <= phi[17] <= 300.148500
55.501000 <= phi[18] <= 302.929000
176.297300 <= phi[19] <= 184.395100

Elapsed Real Time for LP = 3.000 Seconds

LP Predicted Dihedrals	Current Global Minimum
phi[1] = 1.114948 rads* (63.881800 degs)	phi[1] = 1.114948 rads (63.881800 degs)
phi[2] = 3.146228 rads* (180.265600 degs)	phi[2] = 3.146228 rads (180.265600 degs)
phi[3] = 2.847130 rads (163.128538 degs)	phi[3] = 5.212571 rads (298.658300 degs)
phi[4] = 3.074986 rads* (176.183700 degs)	phi[4] = 3.074986 rads (176.183700 degs)
phi[5] = 5.191489 rads* (297.450400 degs)	phi[5] = 5.191489 rads (297.450400 degs)
phi[6] = 3.193061 rads (182.948915 degs)	phi[6] = 5.151067 rads (295.134400 degs)
phi[7] = 2.754424 rads (157.816899 degs)	phi[7] = 3.022198 rads (173.159200 degs)
phi[8] = 5.135328 rads* (294.232600 degs)	phi[8] = 5.135328 rads (294.232600 degs)
phi[9] = 5.124215 rads* (293.595900 degs)	phi[9] = 5.124215 rads (293.595900 degs)
phi[10] = 5.228683 rads (299.581448 degs)	phi[10] = 5.144740 rads (294.771900 degs)
phi[11] = 3.169528 rads (181.600596 degs)	phi[11] = 2.983309 rads (170.931000 degs)
phi[12] = 5.130233 rads* (293.940700 degs)	phi[12] = 5.130233 rads (293.940700 degs)
phi[13] = 5.268207 rads* (301.846000 degs)	phi[13] = 5.268207 rads (301.846000 degs)
phi[14] = 2.996138 rads (171.666080 degs)	phi[14] = 3.172427 rads (181.766700 degs)
phi[15] = 5.266020 rads* (301.720700 degs)	phi[15] = 5.266020 rads (301.720700 degs)
phi[16] = 3.013752 rads (172.675271 degs)	phi[16] = 3.190450 rads (182.799300 degs)
phi[17] = 3.223166 rads* (184.673800 degs)	phi[17] = 3.223166 rads (184.673800 degs)
phi[18] = 3.039320 rads (174.140207 degs)	phi[18] = 1.153593 rads (66.096000 degs)
phi[19] = 3.106989 rads (178.017333 degs)	phi[19] = 3.169434 rads (181.595200 degs)
	Energy = -10.974719 kcal/mol

LP Prediction/Global Minimum Difference (inf_norm) = 2.365441 degs
LP Prediction/Global Minimum Quad Function Difference = 11.392121 kcal/mol

---

Successive LP predicted dihedrals change (inf-norm) = 117.581000 degs

Successive global conformers change (inf-norm) = 238.109400 degs

Energy improvement = 2.248014 kcal/mol

Repeat (y/n)? y

How many randomizations: 60
What fraction of the distance to the global min should be used [0,1]: 0
Use how many points from previous run (global only/all): g

Within what radius of phi[1] = 63.8818 (degs): 5
Within what radius of phi[2] = 180.2656 (degs): 5
Within what radius of phi[3] = 163.1285 (degs): 100
Within what radius of phi[4] = 176.1837 (degs): 5
Within what radius of phi[5] = 297.4504 (degs): 45
Within what radius of phi[6] = 182.9489 (degs): 100
Within what radius of phi[7] = 157.8169 (degs): 15
Within what radius of phi[8] = 294.2326 (degs): 5
Within what radius of phi[9] = 293.5959 (degs): 20
Within what radius of phi[10] = 299.5814 (degs): 5
Within what radius of phi[11] = 181.6006 (degs): 10
Within what radius of phi[12] = 293.9407 (degs): 20
Within what radius of phi[13] = 301.8460 (degs): 5
Within what radius of phi[14] = 171.6661 (degs): 10
Within what radius of phi[15] = 301.7207 (degs): 10
Within what radius of phi[16] = 172.6753 (degs): 10
Within what radius of phi[17] = 184.6738 (degs): 5
Within what radius of phi[18] = 174.1402 (degs): 100
Within what radius of phi[19] = 178.0173 (degs): 5

--> Using 60 randomizations

running QN-BFGS (using only global minimum) ... pass #4

Elapsed Real Time for QN-BFGS = 5461.000 Seconds

--> 62 points reduced to 62 unique conformers

solving LP ...

Conformations: 62 (possibly non-unique)
Dihedral Angles: 19
Quad Fit Error =4.658270 kcal/mol (sum of errors)
LP Pivots = 179

Interpolated Conformations:

1 6 9 12 15 17 21 22 24 30 33 36 38 41 48 49 50 51 55 56 57 59 60 (23 out of 62)

Convex Quadratic Underestimator:

c[0] = +1555.161908
c[1] = -34.543558 d[1] = +0.000000

c[2] = -223.829278 d[2] = +75.446077
c[3] = -0.293176 d[3] = +0.000000
c[4] = -4.439108 d[4] = +0.000000
c[5] = +4.849708 d[5] = +0.000000
c[6] = -2.616177 d[6] = +0.000000
c[7] = +13.611229 d[7] = +0.000000
c[8] = -6.794006 d[8] = +0.000000
c[9] = -14.933655 d[9] = +0.000000
c[10] = +2.805539 d[10] = +0.000000
c[11] = -637.240473 d[11] = +210.926966
c[12] = -19.966918 d[12] = +0.000000
c[13] = -1.045903 d[13] = +0.000000
c[14] = -121.797891 d[14] = +35.933857
c[15] = -13.684770 d[15] = +0.000000
c[16] = +8.283929 d[16] = +0.000000
c[17] = +26.833230 d[17] = +0.000000
c[18] = +0.768751 d[18] = +0.000000
c[19] = +30.518616 d[19] = +0.000000

Max error in Quad Fit = 0.462053 kcal/mol

Distribution of Errors (% of Max)

10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
36	8	4	2	7	1	2	0	1	1

Ranges on the Dihedrals (Degrees):

59.754100 <= phi[1] <= 64.670100
174.363600 <= phi[2] <= 185.634700
58.142400 <= phi[3] <= 299.767500
165.275500 <= phi[4] <= 193.215700
183.900100 <= phi[5] <= 301.015500
52.505800 <= phi[6] <= 303.772100
168.653900 <= phi[7] <= 187.344800
288.604600 <= phi[8] <= 303.332400
290.888000 <= phi[9] <= 300.782500
289.843600 <= phi[10] <= 306.124200
170.113300 <= phi[11] <= 182.814400
289.565600 <= phi[12] <= 306.767000
291.950200 <= phi[13] <= 303.437500
168.734700 <= phi[14] <= 192.165600
291.373000 <= phi[15] <= 307.080400
172.934100 <= phi[16] <= 186.826300
173.068600 <= phi[17] <= 191.041000
57.784600 <= phi[18] <= 300.637400
175.891700 <= phi[19] <= 182.798600

Elapsed Real Time for LP = 4.000 Seconds

LP Predicted Dihedrals Current Global Minimum

phi[1] = 1.114948 rads* (63.881800 degs)	phi[1] = 1.114948 rads (63.881800 degs)
phi[2] = 2.966745 rads (169.981972 degs)	phi[2] = 3.146228 rads (180.265600 degs)
phi[3] = 5.212571 rads* (298.658300 degs)	phi[3] = 5.212571 rads (298.658300 degs)
phi[4] = 3.074986 rads* (176.183700 degs)	phi[4] = 3.074986 rads (176.183700 degs)
phi[5] = 5.191489 rads* (297.450400 degs)	phi[5] = 5.191489 rads (297.450400 degs)
phi[6] = 5.151067 rads* (295.134400 degs)	phi[6] = 5.151067 rads (295.134400 degs)
phi[7] = 3.022198 rads* (173.159200 degs)	phi[7] = 3.022198 rads (173.159200 degs)
phi[8] = 5.135328 rads* (294.232600 degs)	phi[8] = 5.135328 rads (294.232600 degs)
phi[9] = 5.124215 rads* (293.595900 degs)	phi[9] = 5.124215 rads (293.595900 degs)
phi[10] = 5.144740 rads* (294.771900 degs)	phi[10] = 5.144740 rads (294.771900 degs)
phi[11] = 3.021143 rads (173.098729 degs)	phi[11] = 2.983309 rads (170.931000 degs)
phi[12] = 5.130233 rads* (293.940700 degs)	phi[12] = 5.130233 rads (293.940700 degs)
phi[13] = 5.268207 rads* (301.846000 degs)	phi[13] = 5.268207 rads (301.846000 degs)
phi[14] = 3.389502 rads (194.204177 degs)	phi[14] = 3.172427 rads (181.766700 degs)
phi[15] = 5.266020 rads* (301.720700 degs)	phi[15] = 5.266020 rads (301.720700 degs)
phi[16] = 3.190450 rads* (182.799300 degs)	phi[16] = 3.190450 rads (182.799300 degs)
phi[17] = 3.223166 rads* (184.673800 degs)	phi[17] = 3.223166 rads (184.673800 degs)
phi[18] = 1.153593 rads* (66.096000 degs)	phi[18] = 1.153593 rads (66.096000 degs)
phi[19] = 3.169434 rads* (181.595200 degs)	phi[19] = 3.169434 rads (181.595200 degs)

Energy = -10.974719 kcal/mol

LP Prediction/Global Minimum Difference (inf_norm) = 0.217075 degs
LP Prediction/Global Minimum Quad Function Difference = 2.212808 kcal/mol

---

Successive LP predicted dihedrals change (inf-norm) = 135.529800 degs

Successive global conformers change (inf-norm) = 0.000000 degs

Energy improvement = 0.000000 kcal/mol

Repeat (y/n)? n
Run Complete...

F. Q´ as a Negative Definite Matrix

As stated in Chapter VIII (Section C) if $Q' = Q - \mu I$, where $\mu = 1 + \lambda_{max}$ and λ_{max} is the maximum eigenvalue of Q, then Q' is negative definite. This can be shown as follows: Let $\bar{\lambda}$ and $\bar{v}$ be any eigenvalue / eigenvector pair for Q, i.e.

$$Q\bar{v} = \bar{\lambda}\bar{v} \qquad \text{Equation 12.1}$$

Since $Q' = Q - \mu I$, then

$$Q'\bar{v} = (Q - \mu I)\,\bar{v}$$

$$= (Q - (1 + \lambda_{max}) I)\,\bar{v} = (Q\bar{v} - (1 + \lambda_{max}) I\bar{v})$$

Substituting Equation 12.1,

$$Q'\bar{v} = \left(\bar{\lambda}\bar{v} - (1 + \lambda_{max}) I\bar{v}\right) = \bar{\lambda}\bar{v} - \bar{v} - \lambda_{max}\bar{v}$$

$$= \left(\left(\bar{\lambda} - \lambda_{max}\right)\bar{v} - \bar{v}\right) = \left(\bar{\lambda} - \lambda_{max} - 1\right)\bar{v}$$

since $\bar{\lambda} \le \lambda_{max}$, $\left(\bar{\lambda} - \lambda_{max} - 1\right) \le -1$. Thus all eigenvectors of Q are also eigenvectors of Q' and all eigenvalues of Q' are negative. Hence, the matrix Q' is negative definite. ❑

END
10-94